

NASA Conference Publication 3296

1995 Goddard Conference on Space Applications of Artificial Intelligence and Emerging Information Technologies

Edited by
Carl F. Hostetter
*Goddard Space Flight Center
Greenbelt, Maryland*

*Proceedings of a conference held at
NASA Goddard Space Flight Center
Greenbelt, Maryland
May 9 - 11, 1995*



National Aeronautics and
Space Administration

Scientific and Technical
Information Branch

1995

This publication is available from the NASA Center for AeroSpace Information,
800 Elkridge Landing Road, Linthicum Heights, MD 21090-2934, (301) 621-0390.

Foreword

The tenth annual Goddard Conference on Space Applications of Artificial Intelligence and Emerging Information Technologies is sponsored by the Mission Operations and Data Systems Directorate (Code 500) with the participation of the Earth Sciences Directorate's Space Data and Computing Division (Code 930). This year, we have expanded the scope of the Conference to accommodate emerging information technologies such as software agents in recognition of their links to AI.

The mission of the conference is very much the same as it was ten years ago: to offer a forum for practitioners of AI who are engaged in developing and fielding AI systems directed to space applications. Ten years ago, AI was considered a specialty field, a field some claimed was misnamed. Regardless of the controversy, the use of the techniques being promoted in the "AI" discipline continues to grow, as evidenced by the wide variety of contributions (papers, tutorials, etc.) at this conference.

AI is generally accepted today as a valid discipline, i.e., successfully integrated into mainstream computing. It is not unusual to hear of fielded systems containing embedded AI, expert systems, fuzzy logic, etc. Fuzzy logic, for example, has found its way widely into industrial and consumer applications. AI-based systems are now used routinely at NASA to assist with mission planning, science and mission operations, and flight safety, an impressive technology infusion track record.

The Goddard AI Conference has weathered ten years; during this period this Conference has documented the solid progress made in space applications of AI. Our plans are to continue into 1996; the Call for Papers for the 1996 Conference is included in these Proceedings.

The Chair would like to thank the members of the Conference Committee for their contributions in preparing for this Conference; the quality of the event is directly attributable to their efforts and dedication. Thanks to the diligent efforts of a Committee member, we now have a WWW Home Page for our Conference:

<http://defiant.gsfc.nasa.gov/aiconf/AI-conf-General.html>

The Committee would like to thank the speakers, presenters and authors for their contributions; they are the substance of the Conference. I would also like to acknowledge the NASA Center for AeroSpace Information (CASI) for their contribution of abstracts for inclusion in these Proceedings.

The Committee would like to thank Dale Fahnestock, Director of Code 500, for continually supporting the Conference over the years. It is the vision of Goddard management that has made this Conference possible. The Committee would also like to acknowledge Patricia Lightfoot and William Macoughtry for having had the foresight and resolve to initiate this Conference ten years ago.

David Beyer
Chair

1995 Goddard Conference on Space Applications
of Artificial Intelligence and Emerging
Information Technologies



1995 Goddard Conference on
Space Applications of Artificial Intelligence and
Emerging Information Technologies

Conference Committee

David Beyer, ATSC (Chairman)
Joy Bush, CSC
Carolyn Dent, GSFC
Carl Hostetter, GSFC
Peter Hughes, GSFC
Ed Luczak, CSC
Michael Moore, GSFC
Jide Odubiyi, Loral Aerosys
James Rash, GSFC
Nicholas Short, Jr., GSFC
Walt Truszkowski, GSFC
Steve Weaver, CSC

PRECEDING PAGE BLANK NOT FILMED



Table of Contents

Information Management, Natural Language, and Software Agents

Applying AI Tools to Operational Space Environmental Analysis	3 -1
<i>Mike Krajnak, Lisa Jesse, John Mucks</i>	
An Architecture and Protocol for Communications Satellite Constellations Regarded as Multi-Agent Systems	15 -2
<i>Craig A. Lindley</i>	
Automated Database Design from Natural Language Input	29 -3
<i>Fernando Gomez, Carlos Segami, Carl Delaune</i>	
A User-System Interface Agent	45 -4
<i>Nagi T. Wakim, Sadanand Srivastava, Mehdi Bousaidi, Gin-Hua Goh</i>	
Building an Adaptive Agent to Monitor and Repair the Electrical Power System of an Orbital Satellite	57 -5
<i>Gheorghe Tecuci, Michael R. Hieb, Tomasz Dybala</i>	

Planning, Scheduling, and Control

Limits to Ground Control in Autonomous Spacecraft.....	75 -6
<i>Alfred D. M. Wan, Peter J. Braspenning, Gerard A. W. Vreeswijk</i>	
Integrated Planning and Scheduling for Earth Science Data Processing.....	91 -7
<i>Mark Boddy, Jim White, Robert Goldman, Nick Short, Jr.</i>	
A Sustainable Genetic Algorithm for Satellite Resource Allocation.....	103 -8
<i>R. J. Abbott, M. L. Campbell, W. C. Krenz</i>	
A Rule-Based Shell to Hierarchically Organize HST Observations.....	119 -9
<i>Ashim Bose, Andrew Gerb</i>	
Stable Adaptive Neurocontrollers for Spacecraft and Space Robots	131 -10
<i>Robert M. Sanner</i>	
Fuzzy Logic Techniques for Rendezvous and Docking of Two Geostationary Satellites.....	143 -11
<i>Guillermo Ortega</i>	

PRECEDING PAGE BLANK NOT FILMED

Image Processing and Data Classification

- A Hybrid Genetic Algorithm for Resolving Closely Spaced Objects 155 -¹²
R. J. Abbott, W. E. Lillo, N. Schulenburg

- Improved Reduced-Resolution Satellite Imagery 163 -¹³
James Ellison, Jaime Milstein

- Reconstruction of an Infrared Band of Meteorological Satellite Imagery
with Abductive Networks 179 -¹⁴
Harvey A. Singer, John E. Cockayne, Peter L. Versteegen

System Monitoring and Diagnosis

- Model-Based Monitoring and Diagnosis of a Satellite-Based Instrument ... 195 -¹⁵
André Bos, Jörg Callies, Alain Lefebvre

- Linear and Nonlinear Trending and Prediction for AVHRR Time Series
Data 209 -¹⁶
J. Smid, P. Volf, M. Slama, M. Palus

- A Fuzzy Logic Intelligent Diagnostic System for Spacecraft Integrated
Vehicle Health Management 221 -¹⁷
G. Gordon Wu

- Lessons Learned from the Introduction of Autonomous Monitoring to
the EUVE Science Operations Center 229 -¹⁸
*M. Lewis, F. Girouard, F. Kronberg, P. Ringrose, A. Abedini,
D. Biroscak, T. Morgan, R. F. Malina*

- Comprehension and Retrieval of Failure Cases in Airborne
Observatories 237 -¹⁹
Sergio J. Alvarado, Kenrick J. Mock

- A Path-Oriented Knowledge Representation System: Defusing the
Combinatorial Explosion 253 -²⁰
*Stefan Feyock, Stamos T. Karamouzis, John S. Barry,
Steven L. Smith*

- Real-Time Value-Driven Diagnosis 263 -²¹
Bruce D'Ambrosio

- A Linguistic Geometry for 3D Strategic Planning 279 -²²
Boris Stilman

Appendix: Conference Bibliography 1986-1994

Information Management, Natural Language, and Software Agents

1995120955

N95- 27376

51-62

Applying AI Tools to Operational Space Environmental Analysis

47251

P. 11

Mike Krajnak¹, Lisa Jesse¹, John Mucks²

¹GTE Government Systems, 1450 Academy Park Loop, Colorado Springs, CO 80920

²Rome Laboratory/IRDS, Griffiss AFB, NY 13441

November 25, 1994

Abstract

The U.S. Air Force and National Oceanic Atmospheric Agency (NOAA) space environmental operations centers are facing increasingly complex challenges meeting the needs of their growing user community. These centers provide current space environmental information and short term forecasts of geomagnetic activity. Recent advances in modeling and data access have provided sophisticated tools for making accurate and timely forecasts, but have introduced new problems associated with handling and analyzing large quantities of complex data. AI techniques have been considered as potential solutions to some of these problems. Fielding AI systems has proven more difficult than expected, in part because of operational constraints. Using systems which have been demonstrated successfully in the operational environment will provide a basis for a useful data fusion and analysis capability.

Our approach uses a general purpose AI system already in operational use within the military intelligence community, called the Temporal Analysis System (TAS). TAS is an operational suite of tools supporting data processing, data visualization, historical analysis, situation assessment and predictive analysis. TAS includes expert system tools to analyze incoming events for indications of particular situations

and predicts future activity. The expert system operates on a knowledge base of temporal patterns encoded using a knowledge representation called Temporal Transition Models (TTMs) and an event database maintained by the other TAS tools. The system also includes a robust knowledge acquisition and maintenance tool for creating TTMs using a graphical specification language. The ability to manipulate TTMs in a graphical format gives non-computer specialists an intuitive way of accessing and editing the knowledge base. To support space environmental analyses, we used TAS's ability to define domain specific event analysis abstractions. The prototype system defines events covering reports of natural phenomena such as solar flares, bursts, geomagnetic storms, and five others pertinent to space environmental analysis. With our preliminary event definitions we experimented with TAS's support for temporal pattern analysis using X-ray flare and geomagnetic storm forecasts as case studies. We are currently working on a framework for integrating advanced graphics and space environmental models into this analytical environment.

1.0 Introduction

Since the first discovery of radio emissions from the sun, it has become increasingly apparent that solar activity can have a significant impact on

the operation of communication and space-based systems. As we deploy increasingly sophisticated communication and satellite systems, understanding the impact of solar activity has become a significant factor in the proper operation and protection of these systems. Because of this need, the U.S. Air Force and NOAA have established units charged with monitoring the sun and the space environment and alerting potential customers of dangerous geomagnetic conditions that can effect their systems.

Since their establishment, the Air Force Space Forecast Center (SFC) and NOAA's Space Environmental Service Center (SESC) are facing an increasing challenge as the size and diversity of requirements provided by their user community has grown. Their fundamental problem, analyzing and predicting the properties of the space environment, is still a difficult scientific challenge. Other more traditional problems result from serving a growing number of customers without a corresponding increase in staff size. Supporting modern space systems has added to the challenge by requiring more timely and accurate forecasts. Producing these kinds of forecasts has led the Air Force to embark on an ambitious project of developing a comprehensive set of space environmental specification and forecast models (Schunk et. al., 1992) and to the creation of the NSF sponsored Geosphere Environmental Modeling program. Although these efforts have been initially successful, problems related to operationally handling and analyzing the large quantities of complex data still need to be addressed.

Because these problems are perceived as being structured, but not amenable to algorithmic approaches, Artificial Intelligence (AI) has been a natural choice for researchers attempting to find solutions. Indeed, there is a fairly lengthy history of attempts to introduce AI into space environmental analysis (Joselyn, 1993; Schunk et. al, 1992; Shaw, 1989; Burov et. al., 1980). Previous attempts to use AI have focused on special research areas and have resulted in lim-

ited success in the operational environment. Reasons include difficulty in using and understanding AI based programs in an operational setting. Typically users are reluctant to trust AI methods because of the lack of visibility into the reasoning processes. Another problem is that the output normally has to be reformatted to be operationally useful. In addition, users have varying degrees of confidence in the expert who provided the knowledge. Past experience has shown that expert systems are expensive to build and difficult to maintain (Jesse, 1993). While these factors may pose operational problems (Joselyn, 1993; Schunk et. al., 1992) we believe that previous efforts have demonstrated that expert systems represent a sound method for solving some of these analysis problems.

Because of past problems with introducing AI systems into an operational context, we have focused on the operational integration issues. To improve our chances of producing a workable solution, we began development using an AI system that has already been successfully used in an operational environment and is flexible enough to support the analytical tasks we wished to address.

Starting with a basic AI framework we have built a system that performs basic assessments and predictions of the space environment. This framework is extensible so it can be used to address problems related to the introduction of new technology aimed at improving forecaster analyses. We plan to augment our current system with advanced visualization techniques and space environmental models. These methods are aimed at increasing a forecaster's diagnostic and prognostic abilities, however using them can demand more time than a forecaster in an operational environment can afford. Intelligent control of these systems will reduce the analysts' workload and allow them to take advantage of the new insight these methods provide. Integrating these capabilities will provide feedback on the flexibility of the intelligent framework and

an assessment of the effort required to integrate other new technologies.

2.0 Temporal Analysis System

Temporal analysis is a commonly used methodology in the military intelligence environment. Temporal analysis involves the study of events as a function of time to determine patterns of behavior. In this context, an event is a discrete activity that is monitored by the analyst. Aside from a specific type, all events are associated with a time and duration. In this application, events are observations of solar activity and the solar-terrestrial environment. The temporal analysis technique involves displaying the events on a timeline. By displaying historical examples of a particular phenomenon in this manner, analysts are able to establish correlations between observed events and the occurrence of the phenomenon. Once identified, these patterns are recorded and used as a basis for analyzing and new data and making predictions. The practical application of this technique relies heavily on meaningful data fusion and data visualization support.

Because of our focus on operational support issues, we chose to use a general purpose system already in use within the military intelligence community, called the Temporal Analysis System (TAS). The TAS core suite of operational tools supports data processing, data visualization, historical analysis, geographical analysis, situation assessment and predictive analysis. These functions are supported by seven major applications: Timeline, Map, Query Panel, Chalkboard, Dictionary, Model Developer, and Knowledge-Based Predictive Analysis and Situation Assessment (K-PASA). The Map which supports geographical analyses and Chalkboard which supports generic data presentation have so far been omitted from this effort.

TAS has a domain information-based architecture. The database structure and application

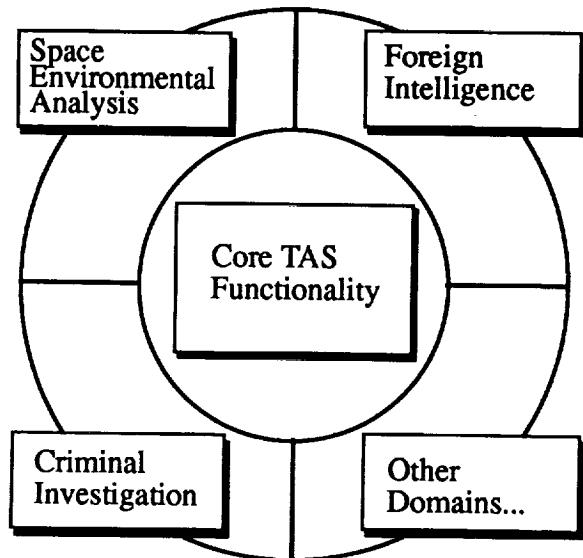


FIGURE 1. TAS Domain Structure.

functionality are separated into general and domain-specific layers. The temporal analysis paradigm provides a broad abstraction around which a significant portion of the system can be built without referring to domain specifics. Support for a particular analysis domain is confined to a separate layer. We often refer to the application specific part of the database and system functions as the domain-dependent layer or simply the "domain". New domains can be layered on top of the core architecture so that new systems can be built reusing 80 to 90% of the core functionality (Figure 1). This approach also has the advantage that functionality developed for one domain is often general enough that it can be promoted to a core capability and shared among the various operational users. The degree of reusability is illustrated by the number of domains currently supported by TAS. These domains include foreign Command, Control and Communications (C3), strategic air, counter-drug, counter terrorism, and criminal investigation.

Data is entered into the database in several ways. The Timeline and Map applications provide basic data entry and maintenance facilities.

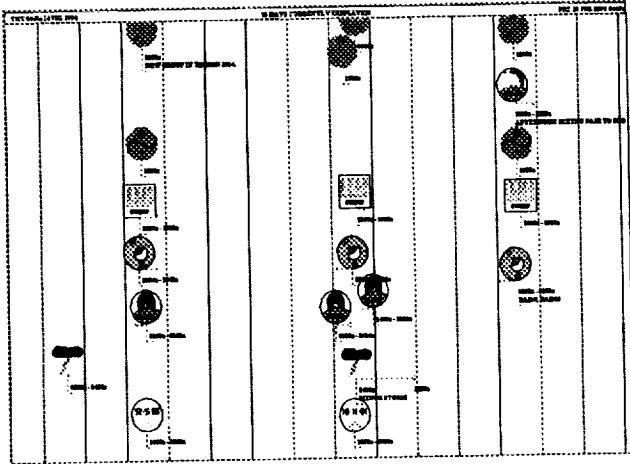


FIGURE 2. Example Timeline Display.

Event data can be loaded manually as well as from real-time message traffic using the local Automatic Message-Handling System (AMHS). Data can also be imported from external historical databases and translated into TAS event specifications.

The Timeline graphically displays events as a function of time (Figure 2). Different event types are identified by icons. For example, in the space environmental domain a radio dish represents radio burst events, a sun with spots represents sunspot report events, and a sun with an eruptive prominence represents general disk and limb activity. Detailed event information can be viewed by clicking an event icon with the mouse. The Timeline supports various data filtering mechanisms designed to aid in the temporal analysis process. Types of filters include event type, icon, keyword, and Area of Interest (AOI). The analyst can customize the Timeline's appearance by changing icon colors, placement, and timescale to create a visually meaningful display. Annotations can be added to communicate additional information such as priority or special significance of a particular event. All of these functions contribute to provide the analyst with an integrated visual summary of complex, multi-source, heterogeneous data.

The Query Panel provides a point and click interface for retrieving data. The user can perform ad hoc queries and have the results piped to external applications for viewing. Displays currently supported are the Timeline, Map, a histogram tool, and a table tool. The Query Panel uses a set of descriptions that provide the attributes and sources that comprise an entity or concept in the users environment. This abstracts the user from the underlying Database Management System (DBMS) and makes it possible to add databases or layer the Query Panel over new databases without modifying the code. The Query Panel graphical interface generates a semi-natural language (SNL) description of the query as it is being built. This capability allows the user to keep track of complex queries and understand their request without having to know the underling data access mechanism (for example, SQL).

The Chalkboard and Dictionary applications are relatively minor. The Chalkboard is a generic drawing tool used to develop briefings. The Dictionary is a user defined lexicon of information. This information includes terminology, definitions and synonym relationships relevant to the specific application domain. Other TAS applications use the Dictionary data to identify keywords and synonyms in incoming data.

The applications discussed so far aid analysts with the manual process of temporal analysis. K-PASA and the Model Developer are expert system tools which help automate temporal analysis by analyzing incoming events for patterns. Model Developer is a knowledge acquisition tool is used to define the knowledge base upon which expert system operates. K-PASA is the engine that compares events against the models stored in the knowledge base to identify situations of interest. The user may select the types of activities that the system should search for among the incoming events. Assessments are displayed in a list ordered by decreasing confidence. The user may select an assessment and

receive either an explanation or a prediction of future activity.

3.0 Knowledge Acquisition & Analysis

Expert systems face a number of special challenges in operational environments. Knowledge may rapidly evolve and require that the knowledge base be constantly maintained. Hard coding systems or systems that require specialized AI knowledge prove to be neither cost effective nor logically practical. Consequently, the knowledge base must be maintainable by a user who works with the system on a day to day basis. This requires a flexible and simple knowledge representation that is easy for users to understand and use.

K-PASA operates on a knowledge base of temporal patterns that are encoded using a knowledge representation called Temporal Transition Models (TTMs or "models") in conjunction with the event database. TTMs are specifications of generalized event patterns that characterize a particular activity of interest. TTMs combine concepts derived from Augmented Transition Networks (ATNs) used in Natural Language Processing (Woods, 1970) and decision trees. Like ATNs, TTMs are composed of states and transitions. States correspond to events in the application area. Transitions describe the temporal relationships between events. States specify the type and characteristics of events which may match the state. For example, a state may specify a type 1B flare that occurs in region 7640. State syntax supports several operators which may be used to constrain event attributes. These operators can be used to define equality, subset or numerical comparison specifications. Temporal constraints can be absolute like "occurs only at noon local time", or can be relative such as "follows in one to ten minutes". Multiple transitions from a particular state are considered a branch. Transitions in a branch are designated as either "AND" or "OR" transitions.

The evaluation of AND/OR transitions is similar to decision trees where OR branches are evaluated independently and AND branches are evaluated together. Each transition has an associated confidence factor assigned by the user. The confidence factor represents the incremental belief that the reported events indicate the phenomenon described in the TTM. Refer to Jesse (1993), for details on the confidence specification and evaluation implementation.

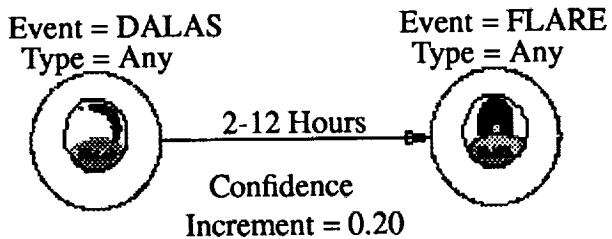
For a simple pedagogical example consider a two state TTM that begins with the observation of disk and limb activity (DALAS) with a transition to an optical solar flare within two to twelve hours (Figure 3A). If no state attribute constraints are in place then the simple existence of a DALAS event satisfies that state. If the system is asked to make a prediction at this point it will only predict the existence of a flare, because in this model the final state is not constrained. For the model to be fully satisfied, a flare event must be detected two to twelve hours after the initial DALAS event. New models can be evolved or updated from existing models. One option for refining this model could be limiting the first event to certain types of DALAS that are more likely to produce flares: more energetic types such as loops, surges, or eruptive prominences (Figure 3B). The final state could also be more specific by constraining the flare to type 1B or greater. K-PASA is capable of evaluating both models.

Associated with TTMs is a graphical specification language developed to be consistent with the manual analysis methods. This language utilizes the same icon notation found in the TAS timeline. The Model Developer implements this graphical language allowing the user to maintain the expert system's knowledge base by means of manipulating the TTMs. The ability to manipulate TTMs in a graphical format gives non-computer specialists an intuitive way of accessing the knowledge base. The ease with which the knowledge base can be created and maintained

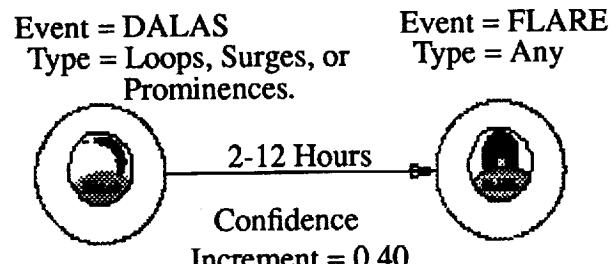
by domain rather than computer specialists has directly contributed to TAS's operational success (Jesse, 1993).

K-PASA performs its assessments by mapping events to TTMs. The core comparison process starts at the TTMs' initial states' specifications. If one or more initial states are satisfied then the system searches for events that satisfy the subsequent transitions and states. This TTM traversal process continues until all TTM branches either terminate or no events satisfy the next transition/state specifications.

The TTM traversal process uses two techniques to increase the flexibility with which models can be applied and accommodate deviations from expected patterns. Deviations can be expected when critical events go unobserved, unreported or if the full range of behavior for the phenomena is not captured by the model. These tech-



3A - Simple DALAS-FLARE Model:
“Any DALAS activity has a 0.20 confidence of transitioning to a flare.”



3B - Extended DALAS-FLARE Model:
“Any energetic DALAS activity has a 0.40 confidence of transitioning to a flare.”

FIGURE 3. Example TTMs.

niques are partial state activation and non-linear processing.

Partial state activation allows user acceptable deviations within the reported events. The degree of tolerance in partial state activation is defined in the states. Each state attribute specification has an associated activation threshold. The possible thresholds are COMPLETE (exact match), UNKNOWN (unknown values are acceptable but at a lower confidence), and MISMATCH (wrong values are acceptable but at an even lower confidence). The level of state activation is derived from the “completeness” of the fit measured by a weighted average of the degree for which each attribute specification has been satisfied. This average is factored into the overall assessment confidence.

Non-linear traversal provides additional flexibility in processing the overall TTM structure. Instead of strictly adhering to the event sequences specified in the TTM, K-PASA will also search for skipped activity and relax the temporal constraints. Relaxing temporal constraints is performed by expanding the expected timeframe defined by the transitions by user defined temporal variances. These variances are relative to the timeframe for which the events should have occurred. As the temporal variance increases, the confidence in the assessment decreases.

Another type of problem is introduced when data is spread across multiple reports. Sometimes, instead of being entered into the system as a single event, information on about a single occurrence is entered as several events. K-PASA compensates for this problem by searching for and combining events that together satisfy a single state. These multiple events contribute to the creation of a meta-event. K-PASA, during the event mapping process, will aggregate those events in order to satisfy the state. A related problem is that an event may be encapsulated into a larger event. The system

will also parse larger events to find embedded events.

K-PASA is integrated with the Dictionary in order to utilize synonym relationships when comparing events to state specifications. Synonym relationships in the Dictionary define terminology equivalency. Examples include acronyms or alternative spellings. Without this integration the user would have to enter all phrases and their associated synonyms in the state specification, even though they semantically represent the same activity.

K-PASA predictive analysis processing is relatively straight forward. The system predicts future events by looking at states yet to be fulfilled. Paths stemming from the last states matched in the assessment are analyzed using the event(s) matching those last states as time references. The constraints in the state specification provide additional information about the predicted event attributes.

In addition to providing analysis capabilities, K-PASA contains an explanation subsystem which justifies system conclusions using a combination of graphics and natural language text (Figure 4). The graphics include a view of the model, with the satisfied states filled, and a view of the timeline that shows only the events which satisfy

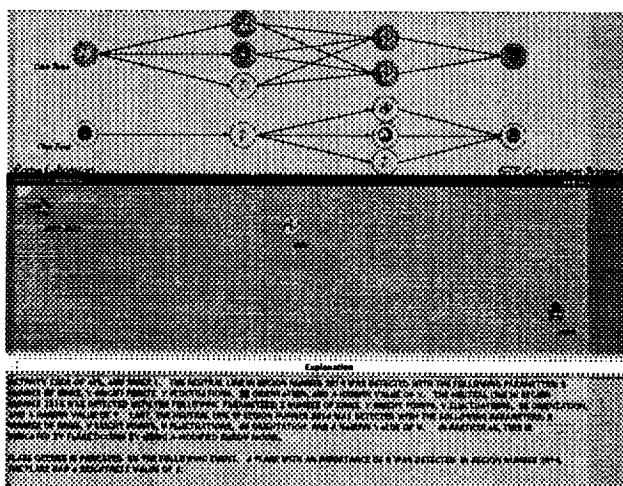


FIGURE 4. Explanation Subsystem

the model. The graphics allow for quick superficial explanation in situations where the analyst is pressed for time. The natural language text provides explanation details when the user has the time and inclination for a more in-depth explanation. Simply reciting the events which matched the TTM is not appropriate. The text must be comprehensive enough to communicate the reasoning behind the assessment without overwhelming the user with irrelevant details. To provide this capability, the text is structured in multiple paragraphs with each paragraph describing the events supporting a particular concept satisfied in the TTMs. The prediction explanation is presented in a similar fashion. The TTM associated with the assessment is shown with the satisfied states filled. States associated with predicted events are highlighted in yellow. The text describes each predicted event along with the expected timeframe of occurrence.

4.0 Space Environmental Domain

Automation of the analytical processes within the forecast centers has been heavily biased towards quantitative methods. These methods include statistical techniques and more recently numerical modeling. While there is a strong agreement that these tools are necessary for improving forecasts, there is some concern that not enough is known about how to properly integrate them into the forecasting process. This stems from the lack of understanding about the physical processes involved and having no well defined analysis model of how the data should be integrated. Human forecasters are able to produce forecasts by working around these problems. They do this largely by applying their experience to determine likely behavior where the quantitative tools cannot be used. This process of predicting results without the use of a mathematical model is known as model-free estimation (Kosko, 1992). Understanding forecasting as a model-free estimation process pro-

vides additional insight into the requirements for intelligent tools.

TAS is especially well suited for the role of supporting qualitative estimation and data integration. As previously shown, TAS focuses on capturing heuristic knowledge. It does not require that a mathematical model be known, but it can use the output of quantitative techniques for analysis. TAS also supports a well defined analysis methodology, which provides a high level framework for systematically integrating various observations. Operational TAS users have reported that the use of the temporal pattern matching methodology closely follows their own reasoning processes when performing event identification, situation assessment, and activity prediction (Jesse, 1993).

For example, in a geomagnetic substorm forecast situation, an analyst might consider current flare activity, the state of the interplanetary magnetic field (IMF), and the configuration of the magnetosphere (e.g. the current dipole tilt, etc.). There are quantitative models which provide at least some degree of insight into these effects, but there is no model which quantitatively describes the relationships. The forecaster accommodates these factors by weighing past experience and considering the similarities and differences in the current pattern. TAS works in conceptually the same manner. From the discussion above, we could build a simple four state model that integrates flare event data such as indicators of the IMF (e.g. solar sector boundary crossings) and magnetospheric indicators (e.g. the time from the last equinox to predict sub-storm activity).

The first step in implementing the space environment domain, after determining its suitability for applying temporal analysis, was to identify the key abstractions or events that would be needed. This process usually requires an ongoing dialog between the software engineers and several domain experts. We utilized a wealth of litera-

ture from the SFC, SESC, and the space physics community and relied on the experience of one of our authors (six years of various space environmental assignments within the military) for our prototype. The framework which abstracts domain specific information from the core functionality allowed easy implementation of the specific event abstractions that we needed. Our first prototype was aimed at building a simple proof-of-concept demo. Extending the prototype will require working with a broader variety of domain experts.

In order to keep the level of effort in line with our goal of only providing an initial proof of concept we narrowed the area of investigation to flare and geomagnetic storm forecasting. Some simple guidelines were established which made the final implementation of the system more useful. For example, we designed event definitions that corresponded to data which could be extracted from real-time message traffic, thus alleviating the need for manual entry of events.

After some initial iterations, we settled on eight event types: BURST, DALAS, FLARE, NEUTRAL LINE, SPOT, STORM, SWEEP, and X-RAY. Table 1 shows these events, a brief description of each, and the message sources from which they can be derived. In order to keep in step with the operational flavor of the work, with one exception, we used the government message formats (USAFETAC/UH-86/003, 1986) to dictate the possible event attributes.

The first step in developing the domain was designing the logical database tables and building the database. The primary database design constraint in the TAS architecture is that the tables must be normalized so that all of the domain independent data resides in a single generic event data table. The index between the domain independent and domain dependent data is a unique event sequence number. After the creation of the database, certain domain depen-

dent portions of the code were modified. These portions were primarily concerned with inserting event data into and extracting it from the database. In addition, new icons were created which helped visually represent the new event types. The actual code changes required about four man-weeks for the initial prototype plus two weeks for testing and refinement.

5.0 Preliminary Experiments

Once the fairly straight forward process of building the domain was complete, we began a series of tests focusing on the ability to bring knowledge into the system and conduct analyses. This primarily consisted of building models, constructing test data, and using K-PASA to compare the test data with the models.

The first set of models were based on fairly simple high level descriptions of possible solar causes for geomagnetic storms (AWS Course 2546-001, 1989; Nishida, 1979; SESC Forecasters Manual, 1989). The basic pattern consisted of a long term precursor (up to a day in advance), an energetic event, detection by satellite sensors, then followed by a storm. For example one of the models consisted of an initial DALAS event that was constrained to be one of the more energetic types. It was followed by an

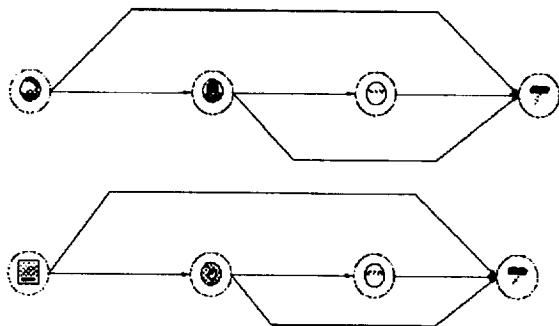


FIGURE 5. Sample Preliminary Models.

optical flare event after a 0 to 1 hour transition. The flare was followed by a GOES x-ray report after a 0 to 3 hour transition and then a storm event after another 1 to 6 hours. Additional transitions allowed for sequences that bypassed one or two of the initial states. Alternatively, the non-linear processing could have been used to handle such cases. Several similar models were built with different constraints, event types (SWEEPS instead of DALAS etc.) and transition values (Figure 5). In conjunction with an artificial set of test data, these first models simply validated the ability to create and evaluate models.

The next set of models, captured more detailed behavior and could realistically be compared to actual data. These models were based upon a

TABLE 1. Space Domain Event Types.

Event	Message Sources	Description
BURST	SEON BURST messages.	Solar discrete radio burst information.
DALAS	SEON DALAS messages.	Disk and limb activity summary reports.
FLARE	SEON FLARE messages.	Solar optical flare information.
NEUTRAL LINE	SESC neutral line analysis charts.	Orientation, and special characteristics of the solar neutral line.
SPOT	SEON SPOTS messages.	Sun spot characteristics, by region.
STORM	SESC/SFC STORM messages.	Geomagnetic storm information.
SWEEP	SEON SWEEP messages.	Solar swept frequency radio burst information.
X-RAY	GOES X-RAY messages.	X-ray measurements from the GOES satellites.

paper by Burov (Burov et. al., 1980) as cited by Sawyer (Sawyer et. al., 1986). Burov's paper extrapolated rules that could be used by a generic rule-based system using a cluster analysis technique with archived x-ray flare data. Burov's rules mixed negative logic (A flare will not occur if...) and positive logic (A flare will occur if...). Since TAS is event oriented and is not geared towards making an assessment driven by the absence of events, the first step was to invert the negative logic. This process was performed using a semi-analytical method that utilized basic symbolic logic. This method compensated for some of the vagaries of the English language and ensured global consistency as individual rules were modified. During the process of defining a TTM for the Burov rules, the use of the Model Developer had a number of advantages as a documentation tool. The ease of use and the clarity of the TTM graphical specification language resulted in an unambiguous and easy-to-follow representation of the knowledge. Evaluation of the Burov rules required more data than the seven events could supply. One or two of these were ignored, based on the premise that they represented rare special cases, or on belief that the data would not be available in an operational environment. For one or two others, reasonable proxies that were available from the current event attributes were substituted. However, since the Burov rules used neutral line characteristics in several ways, this necessitated the addition of a NEUTRAL LINE event. Fortunately this analysis is fairly easy to perform and should only be required to be entered by a user once a day.

The final results were documented as four models. The model representation appears on inspection to capture all of the salient points of the Burov rules. The model representation also has several advantages. As mentioned above, the graphical displays provide a powerful method for documenting the process encapsulated in the model. Also, in conjunction with K-PASA, the TTMs can help the analyst by provid-

ing intermediate assessments of the confidence that a flare will occur. Since the Burov rules do not associate a quantitative value to individual steps, the transition confidences were approximated and will be refined later by comparisons with real data.

6.0 Future Work

We are currently developing a framework for integrating advanced graphics and space environmental models into this analytical environment. This framework will be based on an extended decision support architecture with a central information manager. This intelligent system will configure and execute the appropriate subsystems, as necessary, to support analyst tasks. Examples of potential subsystems include data formatting modules, visualization displays, environmental models, and report generation tools. The planning process will be knowledge-based and utilize criteria such as the forecast product development steps, subsystem execution requirements, and current operational status. Preliminary analysis has indicated that case-based reasoning (CBR) techniques are a viable approach.

As users evaluate the system, additional modifications to the existing prototype will be required. Existing event types will require modification and new ones added to the system. This process can be accelerated by training analysts on the knowledge specification tool, allowing them to construct models, and validating those models with operational data.

Additionally, we plan to reconfigure the TAS AMHS to accept the message formats needed to experiment with the real-time mode. Other enhancements will require precisely defining an inter-process interface to K-PASA so that the new capabilities can be added such as the environmental models.

7.0 Summary

As with many other fields, space environmental forecasters are facing a potentially overwhelming information overload. AI techniques can be utilized to mitigate the problems associated with handling and analyzing large quantities of complex data. AI tools, such as TAS, that are operational in other areas have the potential to solve some of the problems. Whether TAS can be utilized in an space environment operational setting remains to be seen. However, its demonstrated successes elsewhere indicate this approach will prove sound. Once this method of AI assistance is determined to be valid, we hope to expand the framework to include various other data visualization techniques and space environmental models.

References

- Air Weather Service Course 2546-001, *Space Environmental Forecaster Course*, Air Force Space Forecast Center, 1989.
- Burov, V., J. Hirman, and W. Flowers, Forecasting Flare Activity by Pattern Recognition Technique, pp. C235-241, in *Solar Terrestrial Predictions Proc.*, Vol 3, *Solar Activity Predictions*, edited by R.F. Donnelly, U.S. Dept. Govt. Printing Office, Washington D.C., 1980.
- Jesse, L., Temporal Analysis System: A Comprehensive Methodology for Analysis of C³ Intelligence, *1994 Symposium on Command and Control Research and Decision Aids*, Monterey, CA, 1994.
- Jesse, L., Temporal Transition Models: *A New Approach to Situation Assessment and Predictive Analysis*, M.S. Thesis, University of Colorado, 1993.
- Joselyn, J. A., Space Environment Operations Centers: How Can AI Help?, pp. 45-50, in *Proc. of the International Workshop on Artificial Intelligence Applications in Solar-Terrestrial Physics*, edited by J. Joselyn, H. Lundstedt, and J. Trolinger, Lund, Sweden, 1993.
- Kosko, B., *Neural Networks and Fuzzy Systems*, 449 pp., Prentice-Hall Inc., Englewood Cliffs, NJ, 1992.
- McPherron, R. L., Possible Applications of Expert Systems and Fuzzy Logic in Solar-Terrestrial Physics, pp. 1-12, in *Proc. of the International Workshop on Artificial Intelligence Applications in Solar-Terrestrial Physics*, edited by J. Joselyn, H. Lundstedt, and J. Trolinger, Lund, Sweden, 1993.
- Nishida, *Geomagnetic Diagnosis of the Magnetosphere*, Springer-Verlag, NY, 1979.
- Sawyer, C., J. Warwick, and J. Dennet, *Solar Flare Prediction*, 179 pp., Colorado Associated University Press, Boulder CO, 1986.
- Schunk, R. W., R. L. McPherron, J. J. Sojka, W. J. Raitt, and V. B. Wickwar, *Pilot Study of the Feasibility of Using Artificial Intelligence Techniques to Coordinate Specification Models and Real-Time Data for Space Weather Prediction*, 177 pp., Phillips Laboratory Space Physics Division, Hanscom AFB, MA, 1992.
- SESC Forecasters Manual, NOAA Space Environmental Laboratory, Boulder, CO, 1989.
- Shaw, D., THEOPHRASTUS, pp. 6-17, in *Proc. of Fourth Annual Rocky Mountain Conference on Artificial Intelligence*, edited by J. Alexander, Denver CO, 1989.
- USAFETAC, *Climatological Database Users Handbook*, USAFETAC/UH-86/003, 1986.
- Woods, W. A., Transition Network Grammars for Natural Language Analysis, *Communications of the ACM*, vol. 13, pp. 591-606.



1995120956

N95-27377

An Architecture and Protocol for Communications Satellite Constellations Regarded as Multi-Agent Systems

Craig A. Lindley

CSIRO Division of Information Technology
Locked Bag 17, North Ryde NSW Australia 2113

S2-18

47252

P. 14

Abstract

This paper presents an architecture for satellites regarded as intercommunicating agents. The architecture is based upon a postmodern paradigm of artificial intelligence in which represented knowledge is regarded as text, inference procedures are regarded as social discourse and decision making conventions, and the semantics of representations is grounded in the situated behaviour and activity of agents. A particular protocol is described for agent participation in distributed search and retrieval operations conducted as joint activities.

Keywords: Spacecraft Control, Multi-Agent Systems, Postmodern AI.

Introduction

Previous work has defined a layered competency model for autonomous spacecraft according to the Subsumption Architecture for intelligent robotics (Lindley, 1993a). Detailed designs for a power control competency (Lindley, 1993b) and a competency concerned with emergent planning and scheduling of spacecraft payload operations (Lindley, 1994) have been developed. These competencies define what may be called "survival" and "service" competencies, respectively. Survival level competencies are concerned with ensuring the survival of a satellite as an autonomous system, while service level competencies are concerned with the provision of operational services to ground-based users of the satellite. When satellite services are provided by satellite constellations, the issue arises as to which satellite or group of satellites should provide services in support of particular user requests. For satellites to autonomously decide how to collectively provide particular services it is necessary for them to be able to inter-negotiate about the

availability of resources, the presence of competing demands of differing priorities, and the suitability of different platforms for different tasks. These considerations require a further level of satellite competence that will be called the "social" level. This level allows each satellite to participate in the constellation as a member of a system of agents having collective responsibility for providing users services. The ability to negotiate and participate in joint activities represents a form of "high level" cognitive skill that has traditionally been addressed using world-modelling and deliberative reasoning. However, the work described here begins with a behavioural paradigm, and social level competence is built upon a behavioural foundation. The result is a unified *postmodern* framework for artificial intelligence that integrates a social and hermeneutic conception of knowledge with a behavioural conception of semantics. Within this framework, the essence of intelligence is not representation, but the ability to *use* representations within a social context. The postmodern paradigm extends the advantages of behavioural approaches to high level cognition, unifying the survival, service, and social functions of an autonomous system within a single theoretical model.

In this paper, social-level competencies are explored in relation to a protocol for distributed search and retrieval in a satellite network.

Postmodern Artificial Intelligence

Representation-centred artificial intelligence has focussed upon the representation of declarative knowledge in the codifications created during knowledge engineering. The "traditional" view in AI research is that declarative knowledge exists prior to knowledge engineering, and that the tools, techniques, and languages of

knowledge engineering are adequate to "extract", "acquire", or "model" that knowledge. Represented knowledge is processed by an algorithm implementing logical inference, and this is taken as a general model of thought. This "logicist" approach to AI (Kirsch, 1991) manifests many of the assumptions of modernist philosophy, and hence can be referred to as "modernist AI".

In the context of autonomous agency the logicist/modernist approach has led to the traditional functional decomposition of the robot control problem resulting in a high level "horizontal" processing sequence that begins with sensation, with data and control flow then passing through perception, modelling, planning, task execution, and finally to actuator control (Brooks, 1986). This form arises from a dependency upon world modelling at the heart of intelligence, since it reflects the sequence of operations required to update and use a world model in an embedded agent.

Logicist AI has been successful in a number of areas, such as automating some forms of expert reasoning, but it has produced brittle, inflexible systems and poor performance when applied to autonomous agents (Maes, 1993). Poor performance in real time arises from the use of inference as a computational process. Brittleness and inflexibility are ubiquitous problems for representation-centred AI, fundamentally associated with their reliance upon models of the world and the difficulty or impossibility of providing sufficient knowledge to deal with real conditions of system operation. Alternative approaches, such as behavioural AI, have sought to avoid these difficulties by emphasising the embedded, situated nature of agents, shifting the emphasis away from representation of the world and towards a conception of agents as dynamic systems that converge toward equilibrium states (goals). The situated nature of intelligent agency also calls for an emphasis upon the social and cultural contexts of knowledge and expertise representing a further departure from modernist epistemology.

Postmodern discourse (Waugh, 1992) provides a meta-discourse for AI research in which many of the limitations of modernist AI are found to

be symptomatic of modernism in general. This does not mean that AI research should be abandoned, but suggests a major reappraisal of the expectations placed upon, and role of, AI systems, as well as suggesting alternative models of cognition as a basis for principles of engineering practice. AI researchers can take into account the deep limitations of, and conceptual difficulties with, world representation and reasoning, while nevertheless seeking rigorous principles for the development of useful systems within the instrumentalist terms of reference of engineering science.

Several recent developments in AI research can be integrated into a postmodern approach that differs significantly from a modernist approach. In particular:

- behavioural AI incorporates many postmodern principles concerning the situated nature of cognition, and the deficiencies of reason and representation. It emphasises systems that *act in the world*, providing a pragmatic semantics for systems that engage in "high-level cognition".
- concepts of multi-agent systems and distributed AI incorporate postmodern principles of the cultural situatedness of cognition, and in many cases acknowledge the social nature and function of representation and reasoning. It is in the social sphere that representation and inferential reasoning must be located, as conventions for the intercoordination of behaviour, and hence mechanisms for realising the pragmatic semantics of systems of behavioural agents.
- knowledge engineering must be reconceived as a socially embedded authoring process, aimed at articulating agreed (ie. conventional) codes of practice within particular contexts. The results are available for computational processing, and for the intercoordination of the behaviour of both human and artificial processes.

To accept a postmodern position is to abandon ideas of abstract representation, abstract reasoning, and inferential thought processes as the essential constituents of intelligence. Instead,

being-in-the-world becomes the starting point for the analysis of cognition; that analysis must be carried out in full recognition of the situatedness of the analyst, and situatedness must be used as a general concept in the design and structure of computational artefacts. Formal methods of logical analysis and the rationalist methodology of modernism can be adopted as analytical tools, but the broader agenda of the postmodern approach places these in their proper historico-cultural context as tools for *deconstruction* and for the logical synthesis of social codes (processable by computer) for the coordination of behaviour within broader, non-rational, contexts (or forms of life).

The *constructive interaction theory* of Gammack and Anderson (1990) serves as an illustration of what this can mean in the context of knowledge engineering. According to constructive interaction theory, knowledge engineering is an interactive, conversational process, in which the meaning of represented knowledge structures is determined dynamically in the context of knowledge acquisition. The meaning of representations is established with respect to an encompassing reference frame lying outside the data structures, and this reference frame is ultimately never representable (trying to represent it will be infinitely regressive). The outcome of knowledge elicitation can only be viewed "as a specific agreed representation of shared meaning understood by both knowledge engineer and expert". The elicited knowledge is limited to its context of elicitation, and its relevance and applicability to other contexts go unrepresented. Knowledge engineering is a species of conversation, or a *constructive interaction* involving the skills of a domain expert and the knowledge engineer, in which domain knowledge, knowledge of elicitation techniques, and knowledge representation formalisms "interact to produce an agreed representation of domain facts to be coded prior to use by a third party".

It follows that models of knowledge can be used by autonomous agents as resources, much as textual resources are used in human action and problem solving. The coordination of behaviour via texts requires active text interpretation on the part of behavioural agents, during which the

influence of text on the form(s) of behaviour constitute the behavioural semantics of that text. The intensional and semantic processes underlying natural intelligence are situated processes of behaviour and action. The construction of autonomous intelligent systems should therefore begin with behaviour, and behavioural possibilities bound the semantics of representations.

The codification of meanings as a product of behaviour, and as a mechanism for behavioural coordination and interaction, is an issue of the language-using competence of a system. Natural language processing and understanding, as planning and behavioural control, have been extensively addressed as symbolic reasoning problems in the modernist AI paradigm. The postmodern AI paradigm provides an alternative approach to language activity in which situated, behavioural agency supplies the semantics of the production and consumption of codifications. This paradigm promises a unified approach to language use in which plans are viewed as communications together with rigorously defined decision models, informal text objects, and other language artefacts.

Spacecraft and Network Architecture

The AUSTRALIS-1 spacecraft is currently being designed by an informal consortium of Australian universities (ASRI, 1994). The baseline spacecraft is a 35 cm cube, having an expected mass of less than fifty kilograms. It is intended to operate within an altitude range from five hundred to one thousand kilometres. The spacecraft will carry a near infra-red CCD-based camera system, and telecommunications equipment to support a data store-and-forward communications service. AUSTRALIS-1 has particular requirements for autonomy derived from the need to serve a large number of users over a highly dispersed area, using very cheap and simple ground equipment with minimal centralised control or coordination.

AUSTRALIS-1 users will be able to uplink data files, request data broadcast, request image acquisition, and request data deletion. The commands are stored on board the spacecraft, and function as goals within the payload

planning and scheduling competency of the autonomous control system. In the data store and forward mode, ground stations can uplink a request for the spacecraft to broadcast data immediately, or broadcast to a distant ground station specified in terms of a latitude and longitude or a time. Stored data is held in an onboard database within the Command Management System (CMS) for downlinking to specified destinations. Stored data can be deleted upon explicit user request. In all of these transactions the spacecraft will schedule and execute operations without coordination or mediation by a central ground station or command and control network. That is, user stations will interact directly with the spacecraft.

In this paper, the basic satellite model used is extended to include intersatellite communication links, presented to the control system as a set of virtual channel interfaces. This represents a major increase in the complexity of the satellite communications system, and has a significant impact upon overall satellite system design, bearing upon the power system capacity, attitude control requirements, thermal and structural design, and on-board computing requirements. For present purposes, these implications will not be elaborated, and the extended capability will be treated as an extension supported by the currently defined platform. While this is inadequate in practice, it is appropriate for considering the principles involved in integrating social-level competencies with more basic survival- and service-level competencies.

The realisation of social level skills requires an interagent communication medium. This can be realised by some combination of ground network connections, ground-to-spacecraft connections, and spacecraft-to-spacecraft connections. GEO satellites within the system can provide indirect satellite-to-satellite connectivity between a large number of LEO satellites, possibly providing continuous and complete intersatellite connectivity for the LEO system, but at the cost of introducing substantial signal propagation delays, increasing the cost and complexity of the LEO satellites, and incurring the cost of the GEO satellites. Hence there are four major types of agents in the proposed multiagent system: ground stations,

ground network nodes that may connect to other network nodes and/or ground stations, LEO satellites, and GEO satellites. Assuming basic Transfer Layer, point-to-point communications services between physical neighbours of the network, the communications system has some stable subnetworks (on the ground and between the ground and GEO satellites), and subnet connections to and between individual LEO satellites that vary continuously, creating a larger scale network having a very dynamic subset of connections. The system is an open one in the sense that additional user nodes or satellites may be attached or removed at any time.

Multi-Agent Systems

The critical characteristics of agents in multi-agent systems (MAS) that distinguish them from "intelligent" objects and/or processes are the adoption of the terminology and concepts of teleology and social interaction to design the mechanisms of computational process behaviour and interaction, respectively. It is natural to refer to an autonomous satellite as an "agent", since it has goals attributed to it, and the appropriate autonomous coordination and execution of satellite behaviours is required to satisfy those goals. When a number of satellites are available, and user goals could be satisfied by one or more of the satellites, alone or in cooperation, goals become collective and it is necessary to coordinate the behaviours of a number of elements of the system in order to satisfy them. The language and concepts of social interaction naturally arise in the analysis and design of agent cooperation mechanisms for coordinating behaviour to satisfy collective or system-level goals.

Gasser (1991) describes six basic problems that DAI/MAS systems have begun to address that are inherent to the design and implementation of any system of coordinated problem solvers:

1. How to formulate, describe, decompose, and allocate problems, and how to synthesise results among a group of intelligent agents. Suggested bases for decomposition have included abstraction levels, functional, data, or control dependencies, and interaction

- density. Participation of an agent in a social activity is typically described as a *commitment* to a *joint activity*.
2. How to enable agents to communicate and interact: what communication languages or protocols to use, and what and when to communicate. Major approaches include formalised interaction and negotiation protocols.
 3. How to ensure that agents act coherently in making decisions or taking actions, accommodating the non-local effects of local decisions and avoiding harmful interactions. Major approaches include establishing organisation, improving local awareness and skill, multi-agent planning, abstraction, and resource-directed coherence.
 4. How to enable individual agents to represent and reason about the actions, plans, and knowledge of other agents in order to coordinate processes. Principle approaches include the use of utility theory and game theory to represent rational choice, symbolic models of agent capabilities and roles, belief models, and graph models of organisational relationships.
 5. How to recognise and reconcile disparate viewpoints and conflicting intentions among a collection of agents trying to coordinate their actions. Main approaches include assumption surfacing using automated truth maintenance techniques, parallel falsification and microtheories, partial global planning, knowledgeable mediation, standardisation, and various approaches to negotiation.
 6. How to engineer and construct practical DAI systems; how to design technology platforms and develop methodologies for DAI.
- Most systems have been characterised by (Gasser, 1991):
- the use of common interagent semantics with at most one or two meta- or contextual levels
 - a reliance upon correspondence theories of representation and belief
 - global measures of coherence
 - the individual agent as the unit of analysis and interaction
 - dependence upon closed-system assumptions such as shared and global means of assessing coherent behaviour, some ultimate commensurability of knowledge, or some boundary on the system.
- From a postmodern viewpoint, these problems must be addressed without adopting a modernist view of agent cognition. Gasser (1991) suggests that several principles ought to underly the scientific and conceptual foundations of DAI systems from a social perspective:
1. AI research must set its foundations in ways that treat the existence and interaction of multiple actors as a fundamental principle. This raises the question of how representation, reasoning, problem solving, and action should be conceptualised from the social viewpoint. This requires a shift away from the focus of traditional AI upon the individual actor as the locus of reasoning and knowledge, and the individual proposition as the object of truth and knowing. Many of the concepts that have been basic to AI research (such as problems, knowledge, and facts) are regarded from the social perspective as reifications constructed through joint courses of action.
 2. DAI theory and practice must address the basic tension between the local, situated, and pragmatic character of knowledge and action, and ways in which knowledge and action necessarily implicate multiple contexts. The meaning of a specific message is played out as a set of specific response behaviours; however, the responses may be local or distant along some dimension of distribution. Generality of knowledge requires that it should be transportable across contexts, but it must be possible to reintegrate knowledge into a local and situated context for use.
 3. Shared knowledge is not a matter of several agents knowing the same fact interpreted in

the same way, but is a matter of aligning activities in a coherent way. This means that conflicts amount to conflicting actions rather than logical inconsistencies. The question is one of how mutually aligned and supportive commitments can occur and persist.

4. DAI theory and practice must account for resource-limited activity. Resource allocations are the product of the interactions among agents, and resources serve as a channel for interaction among agents.
5. DAI theory and practice must provide accounts of and mechanisms for handling the problems of *joint qualification* (how to establish a basis for joint actions, given the impossibility of fully specifying the assumptions behind a characterisation of any situation), *representation incommensurability*, and *failure indeterminacy* (identifying the source of, or reasons for, a failure).
6. DAI theory and practice must account for how aggregates of agents can achieve joint courses of action that are robust and continuable despite indeterminate faults, inconsistency, etc. which may occur at any level of the system.

All current approaches to distributed coordination rely on a global perspective at some level (eg. semantics, or communication protocols), and assume that the context of negotiation cannot itself be negotiated.

A Discourse System for Agent Interaction

Any implementation of a distributed artificial intelligence (DAI) or multi-agent system (MAS) must address issues of how to provide the communication channels between system components. A number of tool sets extend basic terrestrial communications services to provide additional DAI/MAS facilities, generally emphasising the provision of platform-independent interagent communications and generic facilities implementing various control models, message routing schemes, task distribution schemes, memory management functions, and planning facilities. Examples of

such toolsets include OIS Semantics (Hewitt, 1991), SOCIAL, MACE, ABE, Agora, Cronus, Contract Net (CNET) (Adler, 1992).

Multi-agent approaches have been used both within the structure of individual spacecraft control systems, and as a model for systems having a number of internetworked (semi-)autonomous components. The agent metaphor is highly appropriate for use in the design and implementation of autonomous functions in space and ground support systems, since these systems already involve distributed, interacting agents in the form of human user, operational staff, and mission/spaceship experts and specialists.

SOCIAL has been used in a prototype distributed system for decision support of ground operations for NASA's space shuttle fleet (Adler, 1992). SAGES (Satellite Autonomy Generic Expert System) is a Rockwell project intended to support the reallocation of some ground segment functions onto the spacecraft, where primary functions (such as planning, scheduling, execution, and analysis) are identified with artificial agent roles within the on-board control system (Raslavicius et al, 1989). UNICORN is a blackboard-based multi-agent prototype for spacecraft autonomy, developed at General Electric, in which functions for fault diagnosis and related mission management operations have been developed, along with quantitative subsystem and environment simulations (Rossomando, 1992). Grant (1992/1992) describes a multi-agent approach to the design of the Columbus User Support Organisation (USO), based upon a Message-Based Architecture (MBA) Testbed; implemented in Smalltalk, MBA combines object-oriented constructs (classes, instances, attributes, methods, and messages) with forward-chaining expert systems and generative knowledge-based planning techniques.

This paper is not concerned with the provision of interagent communication channels within the satellite network, or with the application of a multi-agent metaphor within the architecture of a single satellite. Rather, it is concerned with developing a general model of agent interaction

based upon the postmodern conception of knowledge and intelligence described above.

Interagent cooperation mechanisms require several levels of linguistic competency:

- *transport services* must be present to provide point-to-point connections between agents in the network. For the satellite system, many of these connections are dynamic, being established and disconnected as allowed by the changing topology of the network. For any given network node, transport layer connections may be represented by *virtual channel interfaces*. The creation, maintenance, and disconnection of virtual channels is not considered here.
- particular protocols must be supported. If an agent can participate in message exchange facilitated by a particular protocol, it may be said to have the *exchange competency* required for that protocol.
- agents must be capable of engaging in negotiation to establish a joint activity, during which decisions are made about the participation/non-participation of particular agents, the respective roles of the participants, and the allocation of resources controlled by the participants. An agent capable of engaging in a particular form of negotiation may be said to have the *discourse competency* required for that form of negotiation.
- agents must be able to implement the operations constituting the *semantics* of message exchange; ie. the operations that implement the joint activities established by negotiation. An agent capable of implementing the operations required for a particular role in a particular type of joint activity may be said to have the *behavioural competency* required for that role in that type of joint activity. Behavioural competencies include the basic user-service functions of satellites as individual agents within the system.

An agent may participate in joint activities of a particular type if it shares a communication medium with other participants, and if it has exchange, discourse, and behavioural

competencies required to participate in that type of activity. A type of joint activity and its associated competencies can be referred to as a *subculture*; an agent having competencies required to participate in a subculture can be said to be a *member* of that subculture.

Discourse Control

The mechanisms implementing particular competencies can themselves be treated as negotiable conventions. For example, protocol and discourse management procedures can be represented in a common formal language and distributed to all agents that are to participate in the corresponding subculture. In the DAI Open Information Systems (OIS) described by Hewitt (1991), deduction and representation-based reasoning processes are regarded as *microtheories*, based upon a closed world assumption so that derivations can be checked algorithmically for correctness without having to make any observations of the real world or consult any external information sources. Linguistic competencies can be developed and distributed as microtheories. A particular discursive activity may involve the elaboration of a temporary microtheory in the social sphere, with inferential procedures operating upon that microtheory as a social norm of discourse. Microtheories have important strengths in portability (ie. they can be described as stable inscriptions that can be easily stored, moved, and copied) and the self-contained decidability of derivational correctness.

An agent may be a member of several subcultures if it has the competencies required for those subcultures, for each subculture it may participate in a number of joint activities, and for each joint activity it may be in communication with a number of other participants. Apart from the requirement for the presence of appropriate basic competencies, agent participation in joint activities will be constrained by the availability of input and output communication channels, by the availability of on-board memory for the storage of discourse state definitions, and by the availability of sufficient power to support discourse processing and/or the activation of additional memory and communication channel

resources. The allocation of finite resources to discourse functions must take into account competing resource demands and their relative priorities. Resource demands will most likely vary according to the type of a joint activity, the number of its participants, the roles of its participants, and the state of discourse. Discourse control is therefore specific to particular types of joint activities and roles.

Here the particular example of distributed search and retrieval will be considered. This is a highly desirable set of functions for users of satellite bulletin-board and information services, particularly when system resources can be made available while hiding the (dynamic) network structure.

Distributed Search and Retrieval

Search and retrieval is assumed to take place within the scope of a number of *source documents*. The source documents are each subdivided into *logical text units* (or LTUs), that are the individual targets of retrieval. The novel Distributed Search and Retrieval (DSR) system described here is intended to operate within a client-server environment characterised by:

- access via user interface functions having the following characteristics:
 1. A user will be able to request descriptions of search target types.
 2. search target types will be described to the user in terms of taxonomical categories and associated attributes.
 3. A user will be able to request a list of objects that conform to a particular profile specified according to category and attribute values.
 4. A list of item names will be returned to the user.
 5. The user may select any item on the list, and that item will be retrieved and presented on the interface display.

6. Depending upon the problem-solving context, the satellite network structure may not be visible to the user.

- low-level protocols provide Transport services.
- DSR Client processes provide a general purpose interface between DSR User Interfaces and DSR Search processes. DSR Client processes will generally be located at fixed ground nodes within the network.
- search and retrieval processes are implemented within DSR Servers located on physical network nodes together with their associated document files, a description of local taxonomies with their attributes, and a description of the documents or items belonging to each taxonomical category together with their attribute values. DSR Servers may be located either at ground nodes of the network or within satellites.
- DSR Searcher processes dynamically connect DSR Clients and other DSR Searchers to DSR Servers via the (changing) communications network. The search processes may be located on-board satellites or on ground nodes.

The overall architecture is shown on Figure 1. The DSR Client, the DSR Server, and the DSR Searcher are multi-user services, operating as continuous server processes.

On-board the satellite, the DSR Server is a system resource to be controlled. An on-board DSR Searcher must decide:

- whether to support a request for access to its associated DSR Server
- whether and how to pass DSR search requests on to neighbouring nodes in the network
- whether and how to pass results to a DSR Searcher on another node, irrespectively of whether the results come from its own server or that of another satellite

These control decisions must be integrated within the overall control of the satellite.

Search Scoping Techniques

The issue of *search scoping* includes the issues of how to find nodes (and how many nodes) in a network during search (ie. the issue of *network span*), and what objects to search over at any given node (according to the *search context*). Search scoping is implemented using message and database metainformation.

To determine the *network span* of a search, the DSR Client will attach origin and message identification information, a message output time (ie. a time stamp for when the message is issued), and a timeout specification to each message that it issues. Upon receipt of a message, a DSR Server will first compare the origin and message identifiers of the incoming message with a record of the source and origin

dynamics to calculate how the model will change during a DSR session. This information might then be used by a routing strategy to provide return paths for information that may differ significantly from the original search message paths. However, as the network model increases in complexity, each satellite is increasingly likely to have circumspection problems, problems in keeping the model up to date, and greatly increased on-board computational loads.

A much simpler "reactive" solution is proposed here: message meta-information can include the virtual channel identifiers of the immediate connected neighbours of a given satellite that are the sources or sinks of currently active messages. Network dynamism is reflected at each node by changes in the active set of virtual channels presented to the control system. Any

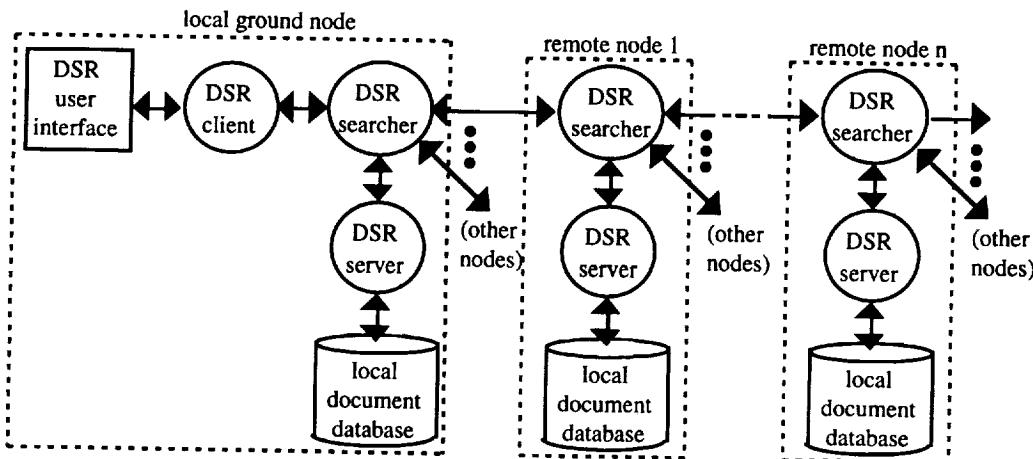


Figure 1. DSR Architecture.

identifiers of messages that it has already processed (stored for their timeout period). If the message is found to have already been processed, it will be ignored. This will eliminate cycles in the network search process. Similarly, the message timeout will be compared with the current system time, and if the timeout period has elapsed the message will be ignored.

Search scoping is critically affected by network dynamism. A sophisticated approach for LEO satellite constellations might be for each participating satellite (ie. agent) to have an up-to-date model of the connection state of the network, and to use knowledge of satellite

message destined for a virtual channel that no longer exists is deleted. This is a very simple scheme (a "flooding" protocol) that results in a DSR scope within a LEO network equivalent to the subnet that exists for long enough for the set of bidirectional message exchanges required by the protocol to be completed, where each DSR Client-server exchange in the session is achieved along a single physical route. Within a satellite constellation containing GEO satellites, the reactive system has a high potential to achieve full network coverage (if message timeout parameters are set high enough). For example, a rule stating that each successive node in a path must belong to a different segment of the

constellation (ie. LEO or GEO) will allow the system to use the stable configuration of GEO satellites to overcome limitations arising from the unstable subnet of LEO satellites. Virtual channel interfaces between LEO satellites and GEO satellites will be highly stable, lasting for one third or more of a LEO orbit period, with smooth and lengthy changeover periods between GEO nodes.¹

The issue of *search context* is the issue of how to improve the relevance of retrieved objects according to the context of the search. Context-dependent searching is implemented using LTU categorisation and search filtering by category and feature constraints. Taxonomical information can be regarded as a form of *self-description* passed between agents, or as a description of the resources controlled by particular agents. In any case, an agent using taxonomical information must be capable of processing that information in order to use it in the specification of a search context. The ability of client processes to process and use those descriptions is critical in the creation of a coherent "society" of computational agents, and requires a higher level of standardisation than that involved in the definition of the DSR protocol described here.

It is possible to classify a single LTU by more than one taxonomy, and all LTUs must be classified by at least one taxonomy. Taxonomies are structures representing metainformation about LTUs, such as position within large-scale text structures (such as traditional books), subject matter, and purpose. It is possible to specify logical conjunctions of categories such that all objects retrieved during a particular transaction will belong to the subset of objects defined by the particular logical expression expressing the logical scope (or document subset) of that search. The specification of constraints upon attributes supports more specific forms of filtering. An abstract taxonomy corresponds to a schema that models object types, object interrelationship types, object attribute types, and subtype/supertype relationships between object types. The

classification and description of a set of LTUs using a taxonomy corresponds to the population of a database with a particular model of instances of the types described by the schema.

A Distributed Search and Retrieval Protocol

Search and retrieval consists of three distinct types of transaction: getting taxonomy descriptions, getting taxonomy items, and getting a selected LTU. This section describes in more detail the message exchange between DSR agents associated with each of these transaction types.

Get Taxonomy Description

DSR Client

- receive a request for local DSR Taxonomy descriptions from the DSR User Interface process
- send a request for local DSR Taxonomy descriptions to the DSR Searcher
- receive a set of local DSR Taxonomy descriptions from the DSR Searcher
- send the set of Taxonomy descriptions to the DSR User Interface process

DSR Server

- receive a request for the local DSR Taxonomy description from the DSR Searcher
- retrieve the local DSR Taxonomy description
- send the local DSR Taxonomy description to the DSR Searcher

DSR Searcher

- receive a request from a DSR Client or Searcher for DSR Taxonomy descriptions.
- send the Taxonomy description request to the local DSR Server
- send the Taxonomy description request to DSR Searchers located at all neighbour nodes other than the immediately preceding node in the path traversed by the request
- receive a Taxonomy description from a DSR Server or Searcher
- send the Taxonomy description to the specified DSR Client or Searcher

¹A detailed analysis of the performance of different network configurations and different reactive routing rules is beyond the scope of this work.

Get Taxonomy Items

DSR Client

- receive a request for a list of items (LTUs) belonging to a specified taxonomical category, or a set of categories, from the DSR User Interface process
- send the request for a list of local items belonging to a specified taxonomical category, or a set of categories, to the DSR Searcher
- receive the list of items belonging to the specified specified taxonomical category, or a set of categories, from the DSR Searcher
- send the list of items to the DSR User Interface process

DSR Server

- receive a request from the DSR Searcher for a list of items (LTUs) belonging to a local DSR Taxonomy category
- retrieve the list of items belonging to the specified category
- send the list of items to the DSR Searcher

DSR Searcher

- receive a request from a DSR Client or Searcher for a list of items belonging to a DSR Taxonomy category of a specified DSR Server
- send the list request to the specified DSR Server or the next Searchers en route
- receive a list of items belonging to a local DSR Taxonomy category from a DSR Server or Searcher
- send the list to the specified DSR Client or Searcher

Get LTU

DSR Client

- receive an LTU specification from the DSR User Interface process
- send request for LTU to the DSR Searcher
- receive specified LTU from the DSR Searcher
- send the LTU to the DSR User Interface process

DSR Server

- receive request for LTU from the DSR Searcher
- retrieve the specified LTU
- send the LTU to the DSR Searcher

DSR Searcher

- receive an LTU retrieval command from a DSR Client or Searcher
- send the retrieval command to the specified DSR Server or the next Searcher en route
- receive an LTU from a DSR Server or Searcher
- send the LTU to the specified DSR Client or Searcher

Message Routing and Scope of Joint Activity

All message transmission associated with this protocol must satisfy the conditions that: the current time from origin for any given message is less than a timeout period specified within the message header, the next virtual channel that a message is destined for must be currently active, and the next virtual channel that a message is destined for must not be its immediate source channel. If either of the first two of these conditions is not satisfied, then a message is deleted instead of being transmitted.

Message routes define the scope of a joint activity, and it is within this scope that a searcher agent decides whether or not to participate in a joint activity. However, the network topology may change during the course of a joint activity. Searcher agents involved in Get Taxonomy Description transactions will define the scope of a joint activity by sending requests to all neighbouring nodes that satisfy the above three conditions. The return paths for these messages define the scope within which local taxonomical descriptions are known to be valid. Between the completion of these transactions and the beginning of new transactions based upon their results (in particular, Get Taxonomy Items requests), some satellites may leave the network while others may join it, depending upon timeout parameters, link bandwidths (ie. data rates), and orbital characteristics. This means that, unless a

transaction completes prior to any such changes, one of the following tactics must be employed:

- taxonomical item retrieval may be initiated with dynamic scoping. Previously acquired taxonomical data may include items that are no longer available, and items may now be available that are not "known" to the initiating DSR Client. Nevertheless, new agents entering the network may be included in retrieval operations.
- new agents are excluded from retrieval operations by using path information to directly address specific DSR Servers. This approach has the advantage of reducing system bandwidth usage, but has the disadvantages that some retrieval operations will not succeed due to agents having left the subnet, and potentially usable information from new agents that have recently entered the potential scope of the joint activity will not be accessible.

A similar tradeoff occurs in relation to the retrieval of a specific LTU: if the LTU is retrieved by explicit route information, network bandwidth is conserved; however, if the network topology changes, the item will not be retrievable, even though it may still be within a dynamically defined scope. Explicit routing is assumed here.

Integration with the Behavioural Spacecraft Control System

The following specifications summarise general discourse rules:

- an agent may be a member of several subcultures
- for each subculture it may participate in a number of joint activities
- for each joint activity it may be in communication with a number of other participants
- agent participation in joint activities will be constrained by:

- the availability of input and output communication channels
 - the availability of on-board memory for the storage of discourse state definitions,
 - the availability of sufficient power to support discourse processing and/or the activation of additional memory and communication channel resources
 - the allocation of resources to discourse functions must take into account competing resource demands and their relative priorities. Resource demands will most likely vary according to the type of a joint activity, the number of its participants, the roles of its participants, and the state of discourse (eg. the negotiation process may require less power than the execution of operations required by an agent once it is committed to participate in a joint activity in a particular role).
 - the current time from origin for any given message must be less than a timeout period specified within the message header, or else the message will be deleted
 - the next virtual channel that a message is destined for must be currently active, or else the message will be deleted
 - a message cannot be sent to its immediate source channel
- These rules must be implemented within the behavioural control system of the spacecraft. It is necessary to support multiple joint activities, roles, and communication links, and to make decisions about resource allocations to requests arising from negotiation processes and lower level survival and service competencies. This can be done using an arbitration mechanism that can assess relative priorities between all of these requests. In general this represents an elaboration of the emergent planning and scheduling system described by Lindley (1994) in which all requests for resources, including those arising from negotiation, are regarded as competing goals. This system is not described in detail here.

Conclusion

This paper has described a distributed search and retrieval (DSR) system for open satellite constellations. The mechanisms for language interaction and negotiation are regarded as social artefacts, with agent behaviours driving the use and application of the language system. Resource allocation to joint activities is achieved by a behavioural control system, and this and the generation of appropriate action sequences (ie. "planning") is achieved in an emergent, non-deliberative, and decentralised way. The system constitutes a distributed multi-agent system that relies upon metaknowledge within the DSR environment to guide the search process and provide search filtering according to the problem solving context. The DSR system is typical of most current multi-agent systems in requiring a priori common objects such as interaction languages, metaconcepts, and behavioural rules or programs to ensure that agents conform to standards. Multi-agent systems as models of human societies require a model of the standard formation process itself, rather than particular standards, to account for the ongoing process of aggregation (as a process, rather than a state) and the fluidity of aggregate boundaries (in terms of knowledge and action) (Gasser, 1993). More fully social computational systems should have the capacity to generate, modify, and codify their own local interaction languages, have degrees of structure and reification that increase and decrease with use, and modify their knowledge and activity structures at all levels of analysis (ie. communities of programs should evolve the languages in which they are written). However, this is not necessary for the successful engineering of systems that depend upon cooperation to achieve openness and flexibility in their functionality, and in this case standards are a necessary prerequisite for the integration of systems into a common facility.

References

Adler R. M. 1992 "Coordinating Complex Problem-Solving Among Distributed Intelligent Agents", *Telematics and Informatics*, 9(3/4), 191-204.

ASRI 1994 *Australis-1 Top-Level Design Document*, Australian Space Research Institute, Sydney, Australia.

Brooks R. A. 1986 "A Robust Layered Control System For A Mobile Robot", *IEEE Journal of Robotics and Automation*, RA-2, no 1, 14-23, March.

Gammack J. G. and Anderson A. 1990 "Constructive interaction in knowledge engineering", *Expert Systems*, Vol. 7, no. 1, Feb..

Gasser L. 1991 "Social conceptions of knowledge and action: DAI foundations and open systems semantics", *Artificial Intelligence*, Vol. 47, 107-138.

Gasser L. 1993 "Social Knowledge and Social Action: Heterogeneity in Practice", *Proceedings, 13th International Joint Conference on Artificial Intelligence*, Morgan Kaufman Pubs. Inc., 751 - 757.

Grant T. J. 1991/1992 "A review of Multi-Agent Systems techniques, with application to Columbus User Support Organisation", *FGCS7*, 413-437.

Hewitt C. 1991 "Open Information Systems Semantics for Distributed Artificial Intelligence", *Artificial Intelligence*, 47, 79-106.

Kirsch D. 1991 "Foundations of AI: the big issues", *Artificial Intelligence*, V. 49, 3-30.

Lindley C. A. 1993a "An Autonomous Satellite Architecture Integrating Deliberative Reasoning and Behavioural Intelligence", *Telematics and Informatics*, 10 (3).

Lindley C. A. 1993b "A Behavioural System for Autonomous Spacecraft Power Distribution and Control", *Artificial Intelligence and Knowledge Based Systems for Space*, 4th Workshop, May 17th - 19th, ESTEC, Noordwijk, The Netherlands.

Lindley C. A. 1994 "On-Board Emergent Scheduling of Autonomous Spacecraft Payload Operations", *9th Goddard Conference on Space Applications of Artificial Intelligence*, May 10-

12, NASA Goddard Space Flight Centre,
Greenbelt, MD, USA.

Maes P. 1993 "Behaviour-Based Artificial Intelligence", *Proceedings of the Second International Conference on Simulation of Adaptive Behaviour*, MIT Press.

Raslavicius L., Gathmann T. P., and Barry J. M. 1989 "An Artificial Intelligence Framework for Satellite Autonomy", *Proceedings, Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Vol 2, 536-543.

Rossomando P. J. 1992 "The Achievement of Spacecraft Autonomy Through The Thematic Application of Multiple Cooperating Intelligent Agents", *Telematics and Informatics*, 9 (3/4), 205-219.

Waugh P. 1992 *A Reader in Postmodernism*, Edward Arnold.

47253

Automated Database Design from Natural Language Input* p. 16

Fernando Gomez

Department of Computer Science, University of Central Florida
Orlando, FL 32816
gomez@cs.ucf.edu (407) 823-2764

Carlos Segami

Dept of Mathematics and Computer Science, Barry University
Miami Shores, FL 33161

Carl Delaune

NASA Kennedy Space Center

Abstract

Users and programmers of small systems typically do not have the skills needed to design a database schema from an English description of a problem. This paper describes a system that automatically designs databases for such small applications from English descriptions provided by end-users. Although the system has been motivated by the space applications at Kennedy Space Center, and portions of it have been designed with that idea in mind, it can be applied to different situations. The system consists of two major components: a natural language understander and a problem-solver. The paper describes briefly the knowledge representation structures constructed by the natural language understander, and, then, explains the problem-solver in detail.

*This research is being funded by NASA-KSC
Contract NAG-10-0120

1 Introduction

In this paper, we describe a system that constructs logical database designs from English sentences entered by users with no knowledge of databases or programming. The logical design used is the entity-relationship model (E-R) (Chen, 1976). The set of user's statements describing a database has been called a *user view* (Navathe and Elmasri, 1986). The techniques for extracting user's views or the relevant components of a logical database from a user are based on elicitation methods. Several methodologies have been developed for aiding the extraction process (Baldissera et al., 1979; Martin, 1981; Ceri, 1983; Albano et al., 1985). More recently, expert systems techniques have been applied to the creation of an E-R model from user's specifications. The VCS system (Storey, 1988) elicits the entities, attributes and relations from the user by asking him/her questions formulated in English. In VCS, the user's replies

are limited to saying "yes/no" and listing the entities, attributes and relations separated by blanks. The approach presented in this paper, however, aims at identifying the entities, attributes and relations from the English descriptions of a problem. For instance, a typical database problem for which our system can build a logical design is the following:

Each person keeps a record of documents of interest. The source and the time of each document are stored with the location of the document. Documents may be books, identified by author name and title. Documents may be also journal articles, identified by journal volume number, author name and title. Documents may be private correspondence, identified by sender and date.

Our system will identify the entities, attributes, key attributes and relations in this passage, and the hierarchical relations between the entity "document" and its sub-concepts "book," "journal article," and "private correspondence." The two main components of our system are a natural language understander (NLU) and a problem-solver. Although research on using natural language processing (NLP) for interfacing databases has been intensive and has achieved certain success (Ballard and Tinkham, 1984; Grosz et al., 1987; Bates et al., 1986), research on the construction of logical databases from natural language has been scarce, but the reader may see (Kersten, 1987; Alshawi, 1985). These earlier attempts are based on syntax. Our approach, however, involves a parse of the sentence, a semantic interpretation of the output produced by the parser, the construction of knowledge representation structures from the logical forms of the sentence,

and the integration of these structures into memory. Figure 1 depicts the main components of the natural language understander module. This model of comprehension of expository texts has been under development for some years now (Gomez, 1985; Gomez and Segami, 1991; Gomez et al., 1993). More recently this model has been applied to the acquisition of knowledge from encyclopedic texts (Gomez et al., 1994) and is the same model being applied here as the front-end to the problem-solver.

The key idea in our approach is to use the final knowledge representation structures as the input to the problem-solver, rather than to use the syntactic output of the parser. The construction of the final knowledge representation structures is done as follows. The semantic interpretation phase, if successful, has built a relation and a set of thematic roles for each sentence. Let us call the thematic roles of the relation the entities for that relation. All the n entities of a n -ary relation are represented as objects in our language, and links are created pointing to the representation of the relation, which is represented as a separate structure called an *a-structure*. For instance, for the sentence *A company sells books of history to customers*, the NLU builds a 3-ary relation with "sell" as the relation and "company," "books of history," and "customers" as entities or arguments of the relation. These three arguments will be represented as separate objects in our representation. The words "relation" and "entities" as used in the preceding paragraph should not be confused with the notions of "relation," and "entities" in the E-R model. Although not identical to the E-R model, this representation is very close to it, making it relatively easy for the problem-solver to decide which entities and relations in our representation stand for entities, attributes and relations in the E-R model.

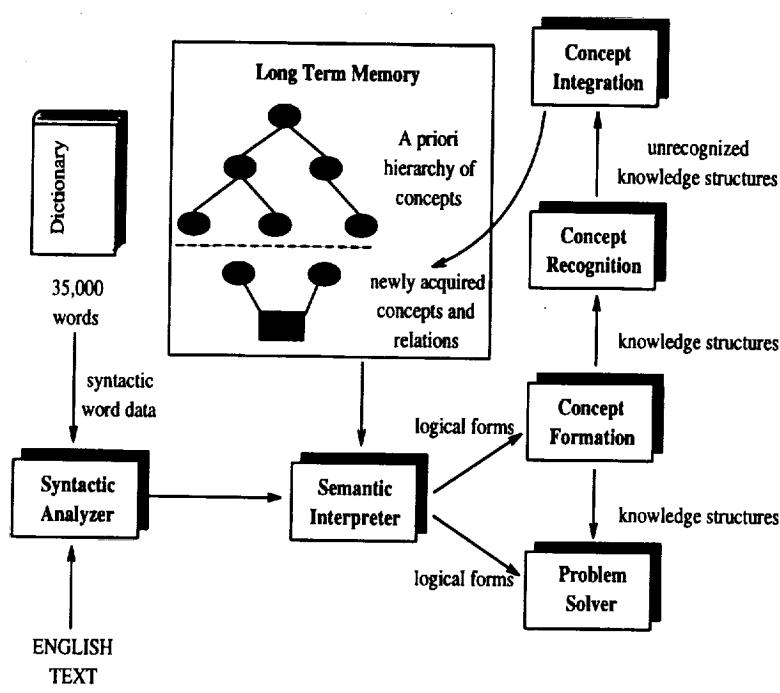


Figure 1: Main Components of the Natural Language Understanding Module

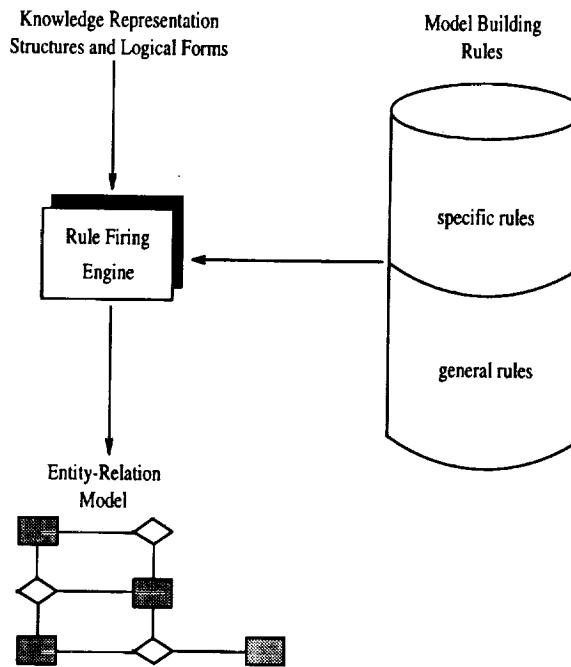


Figure 2: Main Components of the Problem Solver

Figure 2 depicts the main components of the problem-solver. There are two major sources of knowledge used by the problem-solver: the logical database under construction (LDB) and the knowledge structures being built by the NLU. The key idea in the problem-solver has been to decouple the rules that recognize relations, entities and attributes on the basis of the semantics of concepts and the relations built by the NLU from those rules that base their recognition on the entities, attributes and relations already in the LDB. The former rules are called specific rules because they depend on the semantics of the verbs and concepts in the sentence, and the latter are called generic rules because they are independent of the semantics of the sentence.

This paper is organized as follows. The next section describes the knowledge representation structures used by the problem-solver. The remainder of the paper explains the problem-solver, with two major sections describing in detail the specific and generic rules, and their role in constructing an E-R model from the user's sentences. In the last section, we give our conclusions, point out some of the limitations of the system, and future research. An appendix containing an annotated sample session with the system, which is written in Common Lisp and runs on Sparc workstations, ends the paper.

2 Knowledge Representation Structures

Each sentence entered by the user defines one or more conceptual relations. Conceptual relations can have one, two or more arguments, and are classified, accordingly, as unary, binary or n-ary. Also, conceptual relations can be classified as actions or descriptions, depending on the type of their verbal concept.

Nominal concepts that refer to physical or abstract objects are represented as frame-like structures, called *object-structures*. A sample *object-structure* corresponding to the concept "company" is shown below:

```
(company
  (is-a (organization))
  (buy (item ($more (@a6731))))
)
```

The slots in *object-structures* correspond to conceptual relations, which are also represented as frame-like structures, called *a-structures*. In the example above, the second slot corresponds to an instance of the conceptual relation "buy" ("a company purchases items from a number of suppliers"), represented by the *a-structure* @a6731 shown below:

```
(@a6731
  (instance-of (action))
  (args (company) (item) (supplier))
  (pr (buy))
  (actor (company (q (all))))
  (theme (item (q (?))))
  (from-poss (supplier (q (some))))
  (time (present))
)
```

The "args" slot in this structure contains the arguments of the relation, the "pr" slot contains the verbal concept, and the rest of the slots are the semantic cases, also called thematic roles, of the relation. The "q" slot stands for quantifier, and contains the value of the quantifier for that concept. The value of the quantifier may be not only "all" and "some," but also "most," "many," "few," etc. A question mark means that the value of the quantifier is unknown. See (Gomez and Segami, 1991) for a detailed discussion of these quantifiers, and the meaning of these structures expressed in first order predicate calculus (FOPC).

Restrictive modifiers, i.e., complex noun groups, restrictive relative clauses, or nouns modified by prepositional phrases, are represented by an object structure characterized by the presence of a “characteristic features” slot, called a *cf-slot*. The content of the *cf*slot identifies this concept uniquely by providing the necessary and sufficient conditions that define it. The structure is identified by a dummy name (a gensym). Thus, in the sentence *The person who detects the problem writes a problem report*, the restrictive relative clause “the person who detects the problem” is represented by the object structure:

```
(@x5354 (cf (instance-of (person))
    (@a5679)))
```

This structure contains the two characteristic features of the concept: “x5354 instance-of person”, and “x5354 detect problem”. Note that the second characteristic feature is represented by the *a-structure* @a5659 shown below. What appears in the *cf*-slot is simply the name, a gensym, of the *a-structure*.

```
(@a5679
  (instance-of (cf-structure))
  (args (@x5354) (problem))
  (pr (detect))
  (actor (@x5354 (q (constant))))
  (theme (problem (q (?))))
  (time (present)))
)
```

Thus, a *cf*-slot contains one *is-a/instance-of* slot plus one or more names of *a-structures*. We see, then, that the representation of the concepts and relations underlying a sentence consists of a collection of *a-structures* and *object-structures*. As an example, for the sentence *The person who detects the problem writes a problem report*, the NLU builds the following representation structures:

```
(problem-report
  (is-a (report))
  (write%by
    (@x5354 ($more (@a5757))))
)
(@a5679
  (instance-of (cf-structure))
  (args (@x5354) (problem))
  (pr (detect))
  (actor (@x5354 (q (constant))))
  (theme (problem (q (?))))
  (time (present)))
)
(@x5354
  (cf (instance-of (person))
    (@a5679))
  (detect (problem ($more (@a5735))))
  (write
    (problem-report
      ($more (@a5757))))
)
(@a5735
  (instance-of (action))
  (args (@x5354) (problem))
  (pr (detect))
  (actor (@x5354 (q (constant))))
  (theme (problem (q (?))))
  (time (present)))
)
(problem
  (detect%by
    (@x5354 ($more (@a5735))))
)
(@a5757
  (instance-of (action))
  (args (@x5354) (problem-report))
  (pr (write))
  (actor (@x5354 (q (all))))
  (theme (problem-report (q (?))))
  (time (present))))
```

3 The Problem-Solver

The problem-solver is a rule-based system that identifies relations, entities and attributes based on the representation structures built by the NLU and on the current state of the database design. Essentially, the problem solver does its work by accessing the structures that represent conceptual relations, that is, *a-structures*. *Object-structures* are considered only when they define hierarchical relations and in order to access the *a-structures* within cf-slots. The algorithm implemented by the problem-solver consists of two passes. All structures are examined in the first pass, where some structures may result in the creation of database relations, entities or attributes, others structures may cause no action by the problem-solver, and, finally, other may be saved to be considered in the second pass, after the problem-solver has had a chance to gather possibly pertinent information from other structures.

Two distinct sets of rules comprise the problem-solver, generic rules and specific rules. Specific rules are tried first. If they do not succeed, then the generic rules are tried. Specific rules take advantage of the semantic cues in a conceptual relation, when such cues are relevant to the database design. These rules are, therefore, attached to verbal concepts and are fired when the verbal concept in the *a-structure* being considered has rules attached to it. Examples of specific rules are those that construct hierarchical relations among entities, or those that identify key attributes. Generic rules, on the other hand, are fired regardless of the verbal concept in a conceptual relation. They base their actions on the arguments of the relation and on the elements currently

defined in the database design. They are in turn classified as unary, binary and n-ary rules and are applied to unary, binary and n-ary conceptual relations, respectively.

A main driver in the problem-solver controls the order in which the representation structures are examined and the order and kinds of rules that are applied in each case. The first structure examined by the problem solver is always the structure that represents the main clause in the sentence (the main relation). The problem-solver then descends to the structures representing the arguments of the main relation and to explanatory relative clauses, if any. The arguments of the main relation may result in some action by the problem-solver only if their *object-structure* contains a cf-slot, that is, only if the argument is described by a complex noun phrase. In this case, the problem-solver acts on the *a-structures* that are part of the cf-slot of the argument. Let us consider a simple example. Suppose the following two sentences are read:

An organization keeps track of customers, identified by customer id.
The name and address of customers are stored.

The first sentence consists of a main clause and a subclause. The main clause introduces the conceptual relation "organization keep-track-of customer" and the sub-clause, an explanatory relative clause, introduces the relation "customer identified-by customer-id". The problem-solver deals first with the main relation. Since no specific rules are attached to "keep-track-of," a binary generic rule defines "organization" as an entity and "customer," tentatively, as an attribute of "organization." The problem solver then examines the arguments of the main relation, "organization" and "customer." Because both arguments are represented by *object-structures* without cf-slots,

they lead to no action on the part of the problem-solver. Next, the problem-solver examines the conceptual relation introduced by the explanatory relative clause. The verbal concept in this relation has a specific rule attached to it that identifies "customer-id" as a key attribute of "customer." Because "customer" is currently defined as an attribute in the database design, the rule redefines it as an entity, and uses the previous clause to define a database relation. Thus, after the first sentence the problem-solver has built two entities, "organization" and "customer," a key attribute for "customer," "customer-id," and a database relation, "organization keep-track-of customer."

Next, the second sentence is read. This sentence defines a unary conceptual relation. Its verbal concept has specific rules attached to it, which try to identify the arguments of the relation as database attributes. In this case, "name" and "address" are identified as attributes of "customer." We now discuss in detail the different kinds of rules in the problem-solver.

4 Specific Rules

As described above, specific rules are defined for a verbal concept when its semantics indicate that an action specific to the concept must be performed by the problem-solver. Such is the case, for example, with verbal concepts that define hierarchical relations among entities, or those that define key attributes.

Hierarchical Relations Hierarchical relations among entities are introduced by the *is-a* verbal concept. Apart from sentences that explicitly define *is-a* relations, such as *A manager is an employee*, this relation also results from other constructions. For example, for the paragraph:

Each person keeps a record of documents of interest. Documents may be books, identified by author name and title, journal articles, identified by journal volume, number, author name, and title, and private correspondence, identified by sender and date.

the problem-solver creates the entities "document," "book," "journal article," and "private correspondence," and it establishes the conceptual relations:

- "book is-a document"
- "journal article is-a document"
- "private correspondence is-a document"

These conceptual relations are not translated into database relations, but are maintained by the problem-solver to keep track of the inheritance of attributes among entities.

Verbal Concepts that Introduce Attributes

Some verbal concepts strongly suggest that the arguments in the conceptual relation describe attributes of entities. Some specific rules are attached to these verbal concepts in order to identify the attributes and their corresponding entities. The entities may or may not be explicitly identified in the relation. Consider, for example, the sentences: *The source, the time, and the location of each document are stored, The hour and the length of use are recorded, The organization keeps a record of the addresses of the suppliers.* All these sentences are associated with the "store-information" verbal concept, and the arguments in these relations, ("the source of each document," "the time of each document," "the location of each document," "the hour," "the

length of use," "the addresses of the suppliers") all seem to describe attributes of entities. Whether or not they are taken as attributes depends on the current state of the database under construction. If the argument describes a property or characteristic pertaining to a concept that has been defined as an entity by previous statements, then this property is taken as an attribute of the entity. Such is the case, for example, with "the location of the document," if "document" has previously been defined as an entity. If this is the case, then "location" is taken to be an attribute of "document." Thus, after reading the second sentence in the paragraph:

Each person keeps a record of documents of interest. The source, the time, and the location of each document are stored.

the problem-solver identifies "source," "time," and "location" as attributes of "document." In this example, the identification of attributes and entities by the problem-solver is possible because all three arguments of the conceptual relation are represented by *object-structures* with a cf-slot. An examination of the *a-structures* referenced in the cf-slot allows the problem-solver to reach its determination. The same mechanism is used to identify "address" as an attribute of "supplier" from the sentence *The organization keeps a record of the addresses of the suppliers*.

A different situation is illustrated by the sentence *The hour and the length of use are recorded*. Here, the arguments of the relation, "hour" and "length of use," do not explicitly link these possible attributes with any concept, that is, with any previously defined entity. The problem-solver first tries to recognize these concepts by examining the attributes and entities already identified. If this fails, an interaction with the

user is started, in which the problem-solver inquires about entities that might be associated with the arguments of the relation.

A similar mechanism is used for conceptual relations with the "interest-of" verbal concept, such as, "the registration number, the registration termination and the address of a registration office in each state are of interest." The actions taken for this primitive are the same as the actions for "store-information."

Verbal Concepts that Define Key Attributes

Key attributes are typically introduced by the verb "identify" in the passive form, as in *Items are identified by item type*, or *A person, identified by a person id, can own any number of vehicles*. Thus, either the main clause is passive, or it contains an explanatory relative clause in the passive form. Many times, however, key attributes are also introduced by restrictive relative clauses, i.e., *Each vehicle is registered in one or more states identified by state name*. The distinction is important for the problem-solver because the representation structures built for the two cases are different. As we saw above, an argument in a conceptual relation described by a noun restricted by a relative clause is represented by an *object-structure* with a cf-slot. An examination of the *a-structures* in the cf-slot leads us to the "identify by" relation. On the other hand, when an argument in a relation is described by a noun followed by an explanatory relative clause, the representation of the argument does not contain a cf-slot. Instead, the "identify by" relation appears as a conceptual relation in the *object-structure* of the argument. For this reason, after examining the main relation the main driver looks for explanatory relative clauses in the sentence and passes the corresponding *a-structure* to the rule-firing engine.

Other constructions that lead to key attributes result from certain adjectives: *Each major has a unique name*, *Each building in an organization has a different building name*, *The meeting rooms have their own room number*. In all these cases, the verbal concept is “property-r,” and the second argument of the conceptual relation is an instance of the LTM (long-term memory) category “name” (names of things). Thus, a specific rule is attached to “property-r” which examines the representation of the second argument of the relation to verify that it is an instance of “name” modified by the property “unique.” Note that the representation of “unique name” constructed by the NLU is:

```
(@x5476 (cf (is-a (name)) (@a5482)))
(@a5482
  (instance-of (cf-structure))
  (args (@x5476) (unique))
  (pr (property-r))
  (descr-subj (@x5476 (q (all))))
  (descr-obj (unique (q (?)))))
)
```

5 Generic Rules

The second category of rules that comprise the problem-solver are the generic rules. As noted above, these rules are not associated with any particular verbal concept and do their work based only on the arguments of the conceptual relation and the current state of the database design. The steps taken by these rules differ significantly, depending on whether they are unary, binary or n-ary rules. Generally, unary rules result in the definition of attributes; binary rules may define attributes, entities and relations; while n-ary rules result in the definition of database relations. Typically, most sentences in a database description introduce

binary relations. Unary relations normally derive from sentences in the passive form, with verb phrases such as, “are stored,” “are recorded,” “are of interest,” etc.; although we can find sentences like *There are six warehouse locations*.

Unary Rules

When a conceptual relation has a single argument, three cases must be considered: the argument has already been defined as an entity; it has been defined as an attribute; or it does not exist in the database being designed. In each of these cases, the conceptual relation may or may not introduce constraints. These situations are summarized in the following table:

	Argument	Constraint
case1:	Entity	Yes/No
case2:	Attribute	Yes/No
case3:	Does not exist	Yes/No

In the first case, the system interacts with the user to inquire if another entity in the database constitutes a second argument of the conceptual relation. If so, a database relation is created. Otherwise, no action is taken. In the second case, the system inquires if the argument is an attribute of an existing entity. If so, the argument is defined as an attribute of the user-supplied entity. Otherwise, no action is taken. In the third case, if no entities or attributes are currently defined in the database, the argument is defined as an entity. Otherwise, an interaction with the user is started. In all these cases, if the conceptual relation introduces constraints, these constraints are added to the corresponding relation.

Binary Rules

These rules are applied to conceptual relations with two arguments. For each argument the possibilities are: it exists in the database as an entity; it exists as an attribute; or it does not exist in the database.

	Argument 1	Argument 2	Relation	Constraint
case1:	Entity	Entity	Yes	Yes/No
case2:	Entity	Entity	No	Yes/No
case3:	Attribute	Entity	No	Yes/No
case4:	Does not Exist	Entity	No	Yes/No
case5:	Entity	Attribute	No	Yes/No
case6:	Attribute	Attribute	No	Yes/No
case7:	Does not Exist	Attribute	No	Yes/No
case8:	Entity	Does not Exist	No	Yes/No
case9:	Attribute	Does not Exist	No	Yes/No
case10:	Does not Exist	Does not Exist	No	Yes/No

Figure 3: Binary Rule Cases

If the two arguments exist in the database as entities, then the corresponding relation may or may not be defined in the database. These cases are summarized in Figure 3.

The first column indicates whether the first argument of the conceptual relation exists in the database as an entity, attribute, or whether it does not exist. The second column applies similarly to the second argument. The third column indicates whether the relation already exists between the two arguments. The fourth column indicates whether the conceptual relation defines constraints.

In each of these cases the problem-solver defines entities, attributes or relations, updates relations, or adds constraints to a relation. Let us consider case 2. Suppose that "company" and "books" are entities in the database, and that the relation "sell" does not exist in the database. If the user enters the sentence *The company sells books*, the problem-solver defines the database relation "sell" with arguments "company" and "books." Similarly, if the user enters *The company sells books*, and "company" and "books" do not exist in the database design, then the problem-solver creates the en-

tity "company" and defines "book" as an attribute of "company." In each of the ten cases, the actions taken by the problem-solver are the following:

Case 1 Update the relation between argument 1 and argument 2. Some new information may be present in the conceptual relation, such as, quantification.

Case 2 Build a new relation for argument 1 and argument 2.

Case 3 Convert argument 1 into an entity and build a new relation for argument 1 and argument 2.

Case 4 If the relation is *1:n* (meaning the quantifier of argument 1 is 1 and the quantifier of argument 2 is greater than 1), then create a new entity for argument 1 and build a new relation for argument 1 and argument 2. Else, add argument 1 as an attribute of argument 2.

Case 5 Convert argument 2 into an entity and build a new relation for argument 1 and argument 2.

Case 6 Convert both argument 1 and argument 2 into entities and build a new relation for argument 1 and argument 2.

Case 7 If the relation is $1:n$, then create a new entity for argument 1, convert argument 2 into an entity, and build a new relation for argument 1 and argument 2. Else, convert argument 2 into an entity and add argument 1 as an attribute of argument 2.

Case 8 If the relation is $1:n$, then create a new entity for argument 2, and build a new relation for argument 1 and argument 2. Else, add argument 2 as an attribute of argument 1.

Case 9 If the relation is $1:n$, then convert argument 1 into an entity, create a new entity for argument 2, and build a new relation for argument 1 and argument 2. Else, convert argument 1 into an entity, and add argument 2 as an attribute of argument 1.

Case 10 If the relations is $1:n$, then create a new entity for argument 1, create a new entity for argument 2 and build a new relation for argument 1 and argument 2. Else, create a new entity for argument 1, and add argument 2 as an attribute of argument 1.

In each of the previous cases, if the conceptual relation defines some constraints, these constraints are added to the database relation.

N-ary Rules

These rules differ from unary and binary rules in that the final result is always an n-ary database relation for the supplied arguments. The following table contains the possible cases.

	Entities	Relation	Constraints
case1:	Yes	Yes	Yes/No
case2:	Yes	No	Yes/No
case3:	No	No	Yes/No

The first column indicates whether all of the arguments of the conceptual relation exist as entities in the database. The second column indicates whether or not an n-ary database relation exists for the given arguments. The third column indicates whether or not one or more constraints are implied by the conceptual relation.

The problem-solver actions are as follows:

Case 1 Update the relation between the given arguments.

Case 2 Build a new relation for the given arguments.

Case 3 Create a new entity for each of the given arguments which does not have an associated entity in the database and build a new n-ary relation for the given arguments.

As before, in each case if the conceptual relation defines some constraints, these constraints are added to the database relation.

6 Conclusion

We have described an approach to the automated construction of logical database designs for small application domains. The method hinges on using as input for the problem-solver elaborate knowledge representation structures constructed from the logical form of the sentences. Because these structures have a close relation to the representation used in the E-R model, a set of generic rules for the problem-solver can

be systematically derived from these knowledge representation structures and the state of the logical database under construction.

One of the limitations of the system is that, if the NLU is unable to fully interpret a sentence, the problem-solver is not even activated. The user is then asked to rephrase the sentence for which no semantic interpretation was found. But, in many cases there is sufficient information in the partial semantic interpretation for identifying the entities, attributes and relations. In order to make the system more robust, we need to pass whatever partial information the NLU has constructed to the problem-solver. In other words, to make the problem-solver work with less-than-ideal semantic interpretations becomes an imperative for achieving a robust system that does not fail on seemingly easy sentences.

Because the method is based on semantic interpretation, a user needs to convey to the system some background knowledge about the words he/she is using to describe the database application so that the NLU can produce a semantic interpretation. Hence, a major unfinished goal of this work is to design and implement a knowledge acquisition interface by means of which end-users can convey the background knowledge needed by the system to construct a database model for the user's application. We have done some initial investigation of this problem and, in most cases, this is going to require only a mouse click on the part of the user to select one concept among a set of concepts presented by the system. This is possible because the system already operates with a rich ontology of concepts. For instance, suppose that a user wants to write a description of a database including the word "shuttle." He/she will be asked to choose between the three possible meanings of "shuttle": 1) a vehicle to transport things, 2) an instrument when playing badminton, or 3)

a reel. This component is essential for the system to be transported across domains. The goal is to allow the user to tune the system to each specific area of application, without the intervention of programmers, knowledge-engineers or linguists.

The initial goal of this research was to design a problem-solver that would identify relations, entities and attributes with little or no help from the user, and this paper has provided a detailed description of the problem-solver. However, if one brings the user into the loop, the system described in this paper is greatly enhanced. The user can refine the final design of the database by clicking in the entities, attributes and relations. This clicking may result in deleting wrongly identified entities, or attributes, and rearranging some of the entities, relations and attributes. The nature of this interaction will be the object of future reports.

References

- Albano, A., De Antonellis, V., & Di Leva, A. (Eds.) (1985). *Computer-Aided database design*. North-Holland
- Alshawi, H. (1985). Creating relational databases from English texts. In *Proc. of the 2nd IEEE Conference on Artificial Intelligence Applications*, 449-454.
- Ballard, B. and Tinkham, N. (1984). A grammatical framework for transportable natural language processing. *Computational Linguistics*, 10, 2, 81-96.
- Baldisserra, C., Ceri, S., Palegatti, G., & Bracchi, G. (1979). Interactive and formal specification of user's views in data base design. In *Proceedings of the International Conference on Very Large Data Bases*, 262-272.

- Bates, M., Moser, M., & Stallard, D. (1986). The IRUS transportable natural language database interface. In M. Evens (Ed.) *Expert Database Systems*, Benjamin/Cummings, 617-631.
- Ceri, S. (Ed.). (1983) *Methodologies and tools for data base design*. North Holland Publishing Company.
- Chen, P. (1976). The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*, 1,1, 9-36.
- Gomez, F., and Segami, C. (1989). The recognition and integration of concepts in understanding scientific texts. *Journal of Experimental and Theoretical Artificial Intelligence*. 1,51-77.
- Gomez, F., Segami, C., and Hull, R. (1993). Prepositional attachment, prepositional meaning and determination of primitives and thematic Roles. UCF Technical Report.
- Gomez, F., and Segami, C. (1991) Classification-Based reasoning. *IEEE Transactions on Systems, Man and Cybernetics*, 21,3, 644-659.
- Gomez, F., Hull, R., & Segami, C. (1994) Acquiring knowledge from encyclopedic texts. In *Proceedings of the ACL 4th conference on applications of natural language processing*. Morgan and Kaufmann.
- Grosz, B., Appelt, D., Martin, P., & Pereira, F. (1987). TEAM: an experiment in the design of transportable natural-language interfaces. *Artificial Intelligence*, 32, 2, 173-243.
- Kersten, M.L. (1987). A conceptual modelling expert system. In Spaccapietra,
- S. (Ed.). *The entity-relationship approach*. North Holland, 35-47.
- Martin, J. (1981). *An end user's guide for data bases*. Pretince-Hall, Englewood Cliffs, N.J.
- Navathe, S.B., & Elmasri, R., & Larson, J. (1986). Integrating user views in database design. *Computer*, Jan., 50-62.
- Storey, V. (1988). *View Creation*. ICIT Press.

APPENDIX

Sample Session with the Problem Solver

>>> a problem report is written if an anomaly is detected during an operation.

REMARKS: The input sentence is first processed by the NLU, where it goes through the phases of parsing and interpretation, formation of concepts, recognition of concepts, and long-term memory integration of concepts. As an illustration, we show the output of the parser and the representation structures built for the current sentence:

Parser Output:

```

g5301
(subj ((parse ((udt a) (adj problem)
(noun report)))
(ref (indefinite))
(plural nil)
(interp (problem-report
(q (all))))
(semantic-role (theme)))
verb ((aux (is))
(main-verb write written)
(tense pres) (voice passive)
(num sing) (prim (write)))
conj ((if) (interp (if (q (?)))))
sub-cl ((parse ((g5442)))
(sub-clause (g5442)))

```

```
(interp (proposition  
        (q (?))))))
```

```
g5442  
(subj ((parse ((udt an)  
            (noun anomaly)))  
        (ref (indefinite))  
        (plural nil)  
        (interp (anomaly (q (all))))  
        (semantic-role (theme)))  
verb ((aux (is))  
        (main-verb detect detected)  
        (tense pres) (voice passive)  
        (num sing) (prim (detect)))  
prep ((parse (during ((udt an)  
            (noun operation)))  
        (ref (indefinite))  
        (plural nil)  
        (interp (operation (q (?))))  
        (attach-to (verb (strongly)))  
        (semantic-role (at-time))))
```

Representation Structures:

```
(problem-report  
    (is-a (report)))  
(operation  
    (is-a (thing))  
    (related-to (@a5702)))  
(@a5702  
    (args (anomaly) (operation))  
    (pr (detect%by))  
    (theme (anomaly (q (all)))))  
    (at-time (operation (q (?))))  
    (instance-of (proposition))  
    (time (present)))  
(@a5757  
    (args (problem-report))  
    (pr (write%by))  
    (theme (problem-report  
            (q (all)))))  
    (instance-of (action))  
    (time (present)))
```

REMARKS, these structures are passed to the Problem Solver:

PROBLEM SOLVER PASS NUMBER 1

Integrating structure:

```
(@a12754 (args (problem-report))  
        (pr (write%by))  
        (theme (problem-report (q (all)))))  
        (instance-of (action)))
```

firing default-a-structure-delay-integration rule

PROBLEM SOLVER PASS NUMBER 2

Integrating structure:

```
(@a12754 (args (problem-report))  
        (pr (write%by))  
        (theme (problem-report (q (all)))))  
        (instance-of (action)))
```

firing unary-case-3-a rule
creating an entity

Entity: problem-report

REMARKS: Because the structure @a12754 represents a unary relation, the problem solver delays its processing until the second pass. In the second pass, unary rule 3 fires, which defines the entity problem-report.

```
::: next statement  
( 'x' to exit, 'help' to see menu)
```

>>> each problem report is
identified by a unique number.

PROBLEM SOLVER PASS NUMBER 1

Integrating structure:

```
(@a13079 (args (problem-report  
                (@x13008))  
                (pr (identified-by))  
                (descr-subj (problem-report  
                            (q (each))))  
                (descr-obj (@x13008 (q (?))))  
                (instance-of (description))))
```

```
firing prim-implies-key rule  
adding a key attribute to an entity  
Entity: problem-report  
Attribute: @x13008
```

REMARKS: a specific rule identifies @x13008 (unique number) as a key attribute of problem report.

```
::: next statement  
( 'x' to exit, 'help' to see menu)
```

```
>>> each problem report contains  
the name of the person who detected  
the problem.
```

PROBLEM SOLVER PASS NUMBER 1

Integrating structure:
(@a13864 (args (problem-report)
 (@x13792))
 (pr (consist-of))
 (descr-subj (problem-report
 (q (each))))
 (descr-obj (@x13792 (q (?))))
 (instance-of (description)))

firing consist-of-first-arg-entity
rule
adding an attribute to an entity
Entity: problem-report
Attribute: @x13792

REMARKS: a specific rule identifies @x13792 as an attribute of problem-report. @x13792 represents the concept "name of the person who detected the problem."

```
::: next statement  
( 'x' to exit, 'help' to see menu)
```

```
>>> each problem report contains  
the location of the procedure.  
PROBLEM SOLVER PASS NUMBER 1
```

Integrating structure:
(@a14428 (args (problem-report)
 (@x14356))
 (pr (consist-of))
 (descr-subj (problem-report
 (q (each))))
 (descr-obj (@x14356 (q (?))))
 (instance-of (description)))

firing consist-of-first-arg-entity
rule
adding an attribute to an entity
Entity: problem-report
Attribute: @x14356

REMARKS: @x14356 represents the concept "location of the procedure."

```
>>> each problem report contains  
the name of the procedure that was  
being run when the anomaly was  
detected.
```

PROBLEM SOLVER PASS NUMBER 1

Integrating structure:
(@a15516 (args (problem-report)
 (@x15444))
 (pr (consist-of))
 (descr-subj (problem-report
 (q (each))))
 (descr-obj (@x15444 (q (?))))
 (instance-of (description)))

firing consist-of-first-arg-entity
rule
adding an attribute to an entity
Entity: problem-report
Attribute: @x15444

REMARKS: @x15444 represents the concept "name of the procedure"

```
::: next statement  
( 'x' to exit, 'help' to see menu)
```

>>> a problem report is classified as "open" while the problem remains unsolved.

PROBLEM SOLVER PASS NUMBER 1

Integrating structure:

```
(@a16385 (args (problem-report)
                  (open))
              (pr (designate%by))
              (theme (problem-report
                  (q (all))))
              (designation (open (q (constant)))))
              (instance-of (action)))
```

firing binary-case-8-c rule

adding an attribute to an entity

Entity: problem-report

Attribute: open

::: next statement

('x' to exit, 'help' to see menu)

>>> a problem report is classified as "closed" when the problem is solved.

PROBLEM SOLVER PASS NUMBER 1

Integrating structure:

```
(@a17090 (args (problem-report)
                  (closed))
              (pr (designate%by))
              (theme (problem-report
                  (q (all))))
              (designation (closed
                  (q (constant)))))
              (instance-of (action)))
```

firing binary-case-8-c rule

adding an attribute to an entity

Entity: problem-report

Attribute: closed

::: next statement

('x' to exit, 'help' to see menu)

>>> (entities)

The following entities were created:

Entity Name

problem-report

Immediate Parent Entities

Key Attributes

number:

```
0x13008 (cf (is-a number)
           (@a13014))
```

Non-Key Attributes

name:

```
0x13792 (cf (is-a (name))
           (@a13798))
```

location:

```
0x14356 (cf (is-a (location))
           (@a14362))
```

name:

```
0x15444 (cf (is-a (name))
           (@a15450))
```

open

closed

REMARKS: After reading the paragraph, the problem solver has created the entity problem-report, with one key attribute and five other attributes.

54-63

47254

P- 11

A User-System Interface Agent

Nagi T. Wakim
Sadanand Srivastava
Mehdi Bousaidi
Gin-Hua Goh

Center for Research in Distributed Computing
Department of Computer Science
Bowie State University
Bowie, MD 20715

ABSTRACT

Agent-based technologies answer to several challenges posed by additional information processing requirements in today's computing environments. In particular, (1) users desire interaction with computing devices in a mode which is similar to that used between people, (2) the efficiency and successful completion of information processing tasks often require a high-level of expertise in complex and multiple domains, (3) information processing tasks often require handling of large volumes of data and, therefore, continuous and endless processing activities.

The concept of an agent is an attempt to address these new challenges by introducing information processing environments in which (1) users can communicate with a system in a natural way, (2) an agent is a specialist and a self-learner and, therefore, it qualifies to be trusted to perform tasks independent of the human user, and (3) an agent is an entity that is continuously active performing tasks that are either delegated to it or self-imposed.

The work described in this paper focuses on the development of an interface agent for users of a complex information processing environment (IPE). This activity is part of an on-going effort to build a model for developing agent-based information systems. Such systems will be highly applicable to environments which require a high-degree of automation, such as, flight control operations and/or processing of large volumes of data in complex domains, such as, the EOSDIS environment and other multi-disciplinary, scientific data systems.

The concept of an agent as an information processing entity is fully described with emphasis on characteristics of special interest to the User-System Interface Agent (USIA). Issues such as agent "existence" and "qualification" are discussed in this paper. Based on a definition of an agent and its main characteristics, we propose an architecture for the development of interface agents for users of an IPE that is agent-oriented and whose resources are likely to be distributed and heterogeneous in nature.

The architecture of USIA is outlined in two main components: (1) the user interface which is concerned with issues as user dialog and interaction, user modeling, and adaptation to user profile and (2) the system interface part which deals with identification of IPE capabilities, task understanding and feasibility assessment, and task delegation and coordination of assistant agents.

OVERVIEW OF AN AGENT-BASED MODEL

There are almost as many definitions of agents as there are researchers in this field. Traditionally agents have been defined according to their capabilities and architectures. For Miley, "intelligent agents" are nothing but programs "that learn the habits of a user, receive instructions, and then run off to receive or manipulate data" [Miley 1993]. Others, on the other hand, perceive agents as specialized action-oriented entities that can form a collaborative working-group and acquire their knowledge from past experiences available to each other as they collectively attempt to solve a problem [Lashkari 1994]. While Shoham defines his agent as an entity that is perceived to have different states in line with mental components such as belief, capability, choices, and commitments [Shoham 1993]. Lastly, Ted Selker sees an agent as a program "that simulates a human relationship, by doing something that another person could do" [Selker 1994].

However, we define an agent as an entity that is capable of performing information processing tasks which are *delegated* to it with *incomplete specifications*. An

agent may be represented by a processing element, hardware or software, which is *qualified* to perform tasks in a particular *domain*.

Agent Characteristics

An agent can best be described by the following main conceptual and operational characteristics:

- ***Existence***

An agent exists as a processing element. It is created either by initiation or through cloning. A cloned agent inherits the same capabilities and qualification as its parent. However, an agent that is initiated for the first time will evolve to qualify through training.

- ***Self-Determination***

An agent must be able to describe its capabilities to potential users/clients. This property, which is a type of reflection, is essential in order for an agent to determine whether or not to delegate a task to a particular agent.

- ***Delegation***

An agent must be able to accept delegated tasks as well as be able to delegate tasks to other agents. Therefore, an agent may play the role of a client or a server depending on its responsibility in performing a task. However, delegation should occur only after it has been determined that an agent is capable of performing a task.

- ***Operation***

There is a number of capabilities which define the operational aspects of an agent:

(1) *Concurrency*: being able to operate in parallel and, therefore, contributing to an improved system performance.

(2) *Autonomy*: requiring minimum intervention from other agents or users and, therefore, possessing a greater level of independence.

(3) *Cloning*: being able to reproduce itself with identical capabilities and, therefore, maximizing system reliability and performance through dynamic parallelism.

(4) *Migration*: being able to relocate from one node to another in a distributed system. This can lead to improved efficiency through balancing workload, minimizing network communications, and providing locality of service.

(5) *Persistence*: being able to try different possibilities in a solution space until a task is performed provided no time constraint is violated.

- ***Communication***

An agent can communicate with other agents in four different ways:

(1) Direct manipulation takes place when an agent directly instructs another agent to render a service. This is used in support of task delegation.

(2) *Confirmation* is a way for an agent to ask another to confirm an action, usually by responding with *yes* or *no*.

(3) *Feedback* is a way of providing positive or negative reinforcement after the completion of a task. This helps agents assess their own performance and learn from their own experience.

(4) *Negotiation* is a way for two agents to enter into a brief dialog in order to agree on some terms and/or constraints before a task is delegated.

The Qualification and Trust Factor

Since agents are intended to perform complex tasks, mostly independent of the human user, it is essential that an agent be qualified to perform tasks in a particular domain. Therefore, we extend our prior definition of an agent to include *qualification* while recommending that a computer process does not qualify to be an agent unless it meets the following:

- The process's program must be correct. That is, it must conform to design specifications and testing standards based on proven software engineering principles. However, a correct program does not imply that the corresponding agent will be able to perform all the tasks delegated to it.
- The process must have access to a knowledge base within a well defined domain. The knowledge base itself must be correct, that is, its facts and rules are consistent and it has been verified by a (human) domain

expert. However, completeness is not a prerequisite to correctness.

It is important that all of agent's qualification standards be observed, for it to qualify to perform various tasks, and be trusted [Lashkari 1994] by the users or the agents it assists. All agents become qualified after a period of training through which the knowledge base itself is built. Once an agent becomes qualified, it is then ready to assume its responsibilities. Therefore, since an agent is a representative of a user (directly or indirectly) with an opportunity of being delegated tasks, it has to be *trusted* based on certain level of *confidence* which can only be determined through *qualification*.

MULTI-AGENT BASED SYSTEMS

A multi-agent based system is an environment in which a community of agents work collaboratively on solving problems with a common domain. However, each individual agent has a particular role to play which depends on the expertise and the specialization of the agent.

Since agents are highly specialized and are often distributed over a network of computers, it becomes more difficult to provide potential users with transparent access to system services. Therefore, we introduce an interface agent to facilitate such access.

THE USER-SYSTEM INTERFACE AGENT (USIA)

USIA is a special agent that may be thought of as a "middle-man" between human users and an information processing environment (IPE). An

USIA may also be viewed as a front-end system which provides human users with a transparent interface to a community of agents of which each agent may have a different type of expertise and, hence, a special interface protocol. Therefore, without an USIA, a user who is in need of information processing services will need to first locate other agents in the IPE that are capable of performing its task and then learn to interact directly with each of them based on their interface protocols.

USIA offers an intuitive approach to the way a user can request services from an agent-based system by shifting the burden of locating and interacting with agents from the user to itself.

Main Responsibilities of USIA

USIA accommodates interaction with a whole spectrum of users ranging from novices to experts. In doing so, it performs a series of tasks:

- *User Dialog:* USIA provides its users with interaction capabilities through a graphical interface which offers two types of interaction media:
(1) a taxonomy-based 'select-and-combine' type of interface that is dynamically derived from domain-specific services which are available in the IPE and (2) a restricted query language that is simple enough for novice users to state their fuzzy and often ambiguous requests, but is capable enough for expert users to state their specific and often complete requests.
- *User Adaptation:* The main advantage of USIA over a common

interface system is that it is capable of monitoring user interaction with the IPE and, based on user modeling techniques, it is capable of adapting to changes in user profiles. The purpose of user modeling is to give USIA the ability to predict user behavior and, hence, assist the user more efficiently. USIA is also capable of gathering unobtrusively usage patterns and offers facilities to automate them and build a knowledge base of user models. This knowledge base is then used in two ways: (1) to aid in resolving ambiguity in user requests and, hence, understanding them and (2) to predict possible next steps in user requests and, therefore, minimize interaction.

Task Understanding and Delegation: In order for USIA to handle high-level requests for processing information (e.g., data searches), it needs to complete a number of steps: (1) be aware of the capabilities of the IPE as reflected in a knowledge base and the services provided by the Agent Manager, (2) analyze and understand a request, (3) decompose a request into a set of tasks, (4) assess service feasibility based on the current state of IPE capabilities, i.e., availability of agents with the needed specialty, (5) delegate tasks to qualified agents, and (6) coordinate execution and assemble and communicate results back to the user.

A Real-World Analogy of USIA: The Hotel Concierge

One way to model USIA is to think of it as a concierge in a hotel environment whose main role is to assist hotel patrons in obtaining services which are in turn provided by various types of specialists. The pool of resources available to the concierge may include specialists such as car rental agents, travel agents, laundry cleaning agents, and taxi cab dispatcher agents.

As Figure 1 illustrates, a hotel concierge is an interface between a hotel guest (i.e., a user) and the specialist agents (i.e., the IPE). A hotel patron may ask for a variety of services from the concierge. Once the concierge accepts a service request, it then identifies the appropriate specialists and delegate responsibilities to them.

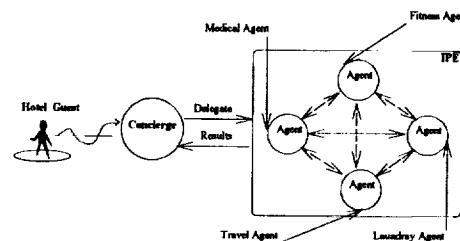


Figure 1. A Real-World Analogy of USIA.

Suppose, for example, that a guest desires to take a vacation somewhere, and would like the concierge to handle all the necessary arrangements, such as air travel, hotel accommodation, and tour guides. All the guest has to do is to present him/herself to the concierge and to ask for the services with the desired specifications, such as, intended date and time, travel destination, and cost range. The guest may also specify any special preferences that he/she might have

concerning choice of an airline, the type of seat, and the type of meal.

The main point here is that the concierge, which is a special type of agent, must be able to provide different types of support and, therefore, handle the following different modes of interaction. However, in all cases, we assume that the service requester (i.e., a hotel patron) always has a goal or a purpose, such as, the intent to take a vacation.

- 1) The user knows the task (i.e., what to request, such as, arrange a vacation to Bermuda for two people during the month of January), knows the task is feasible, has the expertise (i.e., the 'know how'), but needs someone else to perform the task for him.
- 2) The user knows the task, knows the task is feasible, but does not have the expertise to perform the task.
- 3) The user knows the task but has no information on its feasibility.
- 4) The user only has a goal but has no knowledge of what to do, whether or not it is can be done, or how to do it.

Obviously, each type of user requires a different level of attention from the concierge and, therefore, the kind and length of dialog will vary with each type of user.

This example highlights an important role that a user interface agent can play in providing services transparently and efficiently to various types of users

whose requests may require several sources of expertise in order to be serviced.

• *Architectural Highlights*

As illustrated in Figure 2, the architecture of USIA is comprised of two main components: the User Interface (UI), and the System Interface (SI). The User Interface is responsible for facilitating the interaction with human users, monitoring their behavior in order to learn their habits, and be able to adapt in order to better serve their needs. In turn, UI interacts with the System Interface which is responsible for interacting with a community of specialty agents in the IPE in order to process service requests.

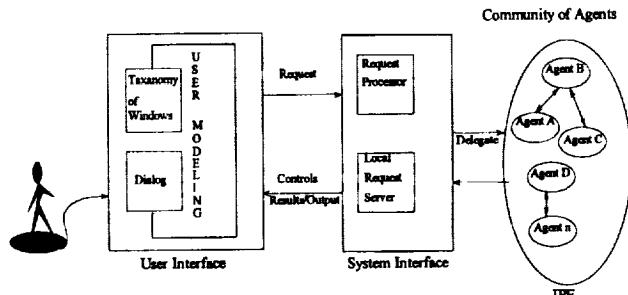


Figure 2. USIA's High Level Architecture.

Figure 3 outlines a detailed architecture of an USIA prototype system which has been developed as a front-end to an agent-based system, known as AFLOAT [Truszkowski 1993], for Report Generation in the Flight Operations domain at the NASA/Goddard Space Flight Center.

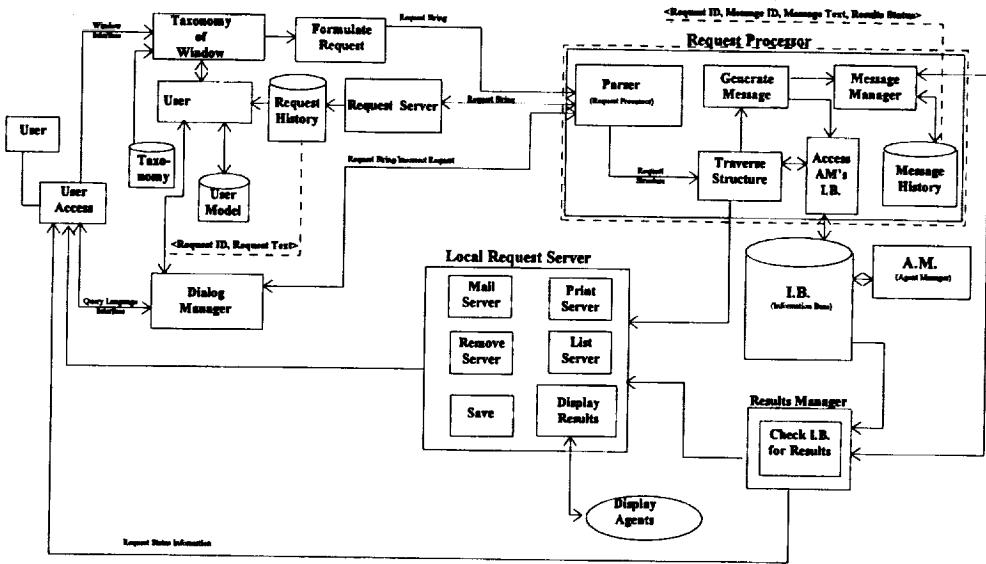


Figure 3. USIA Prototype Architecture.

- ***The User Interface Module***

The primary goal of the UI module is to formulate a user request for information processing services and pass it to the System Interface for processing. In the first phase of development, USIA employs two interaction mechanisms, as shown in Figure 4-a:

1. The user is presented with a dynamically generated, domain-driven taxonomy of windows from which the user ‘selects-and-combines’ services based on which a request statement is formulated by the system. Figure 4-b shows a snapshot of sample windows for the Flight Operations Report Generation domain.
2. The user types in a service request as a query statement chosen from a restricted, intuitive language which

was developed for this domain. This language has capabilities which range from being able to show available services and generate reports to being able to display and mail reports. The following are some examples statements:

- *show category command and data handling subsystems*
- *generate category command and data handling subsystems report orbit decay starting 11/10/84 ending 12/13/84 in graphics*
- *mail report orbit-decay-1 to tom@internet, anne@internet*

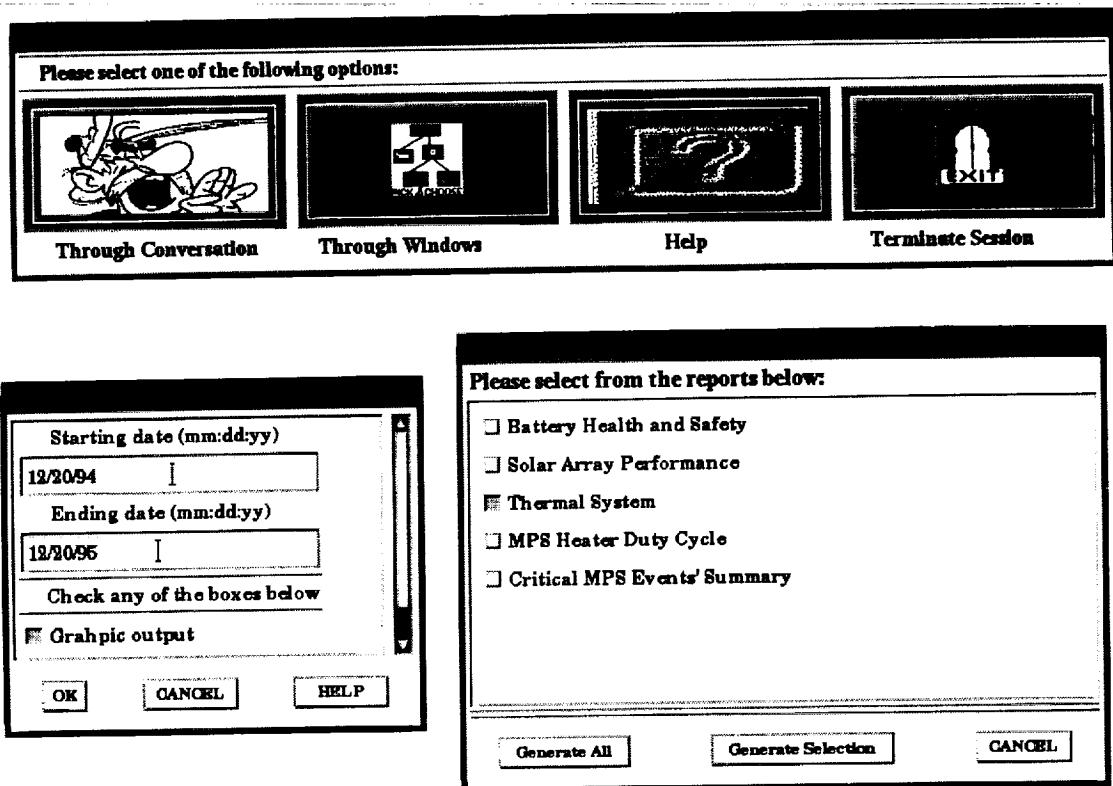


Figure 4-b. Graphical User Interface to a Report Generation and Management Application.

Once a request has been formulated, it is passed on to the System Interface for processing. Upon execution, results are then presented to the user through UI. Figure 5 illustrates the main steps of UI.

This version of USIA incorporates minimal user modeling techniques which include capturing user requests and logging them for comparative analysis in order to predict future user behavior in requesting services. It also allows for automating tasks based on users preferences for routine and off-line processing. However, efforts are underway in the second phase of USIA's development to employ a significant user modeling component which addresses issues such as: (a) modeling of individual users as well as classes of user populations, (b) a structure for user

models, (c) techniques for identifying changes in users behavior and to reflect them in the corresponding models, and (d) methods for adapting to changes in user models in order to serve the end user more efficiently.

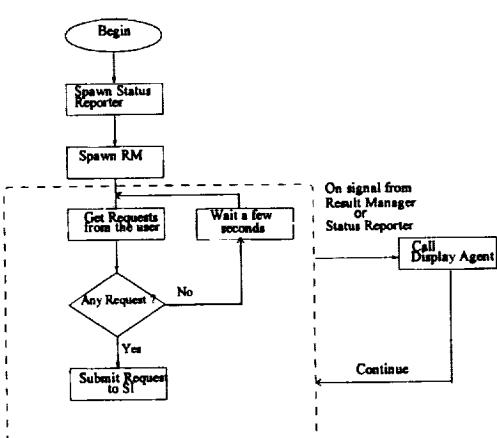


Figure 5. User Interface Flowchart.

- **The System Interface Module**

The goal of the SI module is to process a service request which has been received from a user via the UI module. Each request is first parsed and analyzed for grammatical and semantically correctness. Upon detecting any errors (including ambiguity), USIA attempts to correct the request based on its knowledge of the domain and the user (through user modeling) and may enter in a dialog with the user for request clarification purposes if necessary.

In addition, SI is responsible for decomposing a request into tasks- a *task* is defined as one unit of work which can be delegated to a single agent at one time. Therefore, depending on the available pool of specialty agents, a service request may be decomposed into one or more tasks. Also, a request may be either *local* or *remote*. A local request is one which can be processed by USIA and need not be delegated to another agent, for example: a request to list reports which have been already generated and are saved in the user's work space.

However, a remote request is executed by delegating each of its corresponding tasks to an agent that is capable of performing it. In order for USIA to assess the feasibility of a request, it utilizes an Information Base (IB) which catalogs information on all which are available in the IPE at that particular time. For each agent, the IB stores a list of its skills which is used by USIA in order to determine which, if any, agent is capable of performing a particular task. Upon making such a determination, SI formulates a special message and

delegates the task to the agent while assisted by an Agent Manager (AM), which is responsible for locating the agent and dispatching the message (i.e., the task) to it. In our present configuration, there is one AM for each node of a distributed IPE.

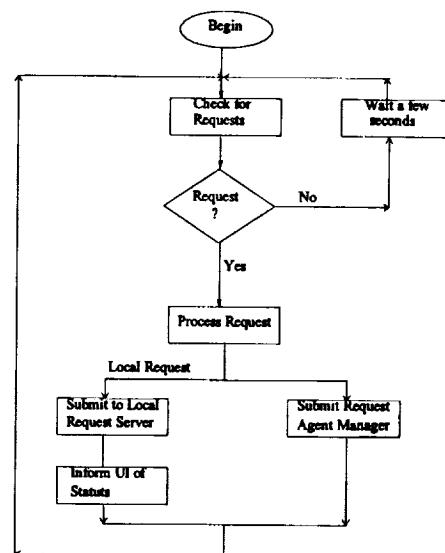


Figure 6. System Interface Flowchart.

Once a task is delegated and performed by an agent, its results are communicated to a Results Manager via the IB. The Results Manager is a special daemon which is responsible for assembling outcomes from processing a request (by executing one or more tasks) and for informing the UI module, which in turn notifies the user. Once results are ready, a user may choose to display them and/or save them. Special agents are utilized depending on the type and format of the result object.

Upon request by user, special agents are utilized to display the results depending on the format and type of a result object.

Integrating UI and SI Modules

Figure 7 illustrates the cyclic flow of high-level activities from the user through the different components of USIA and the interaction with the IPE, via the Agent Manager and the Information Base, and back to the user. We should note that the whole processing of a request is done in the background and transparently from the user.

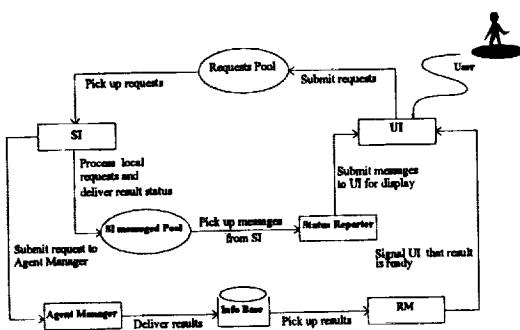


Figure 7: User Interface & System Interface Interaction

CONCLUSION

An Assessment

The first version of USIA demonstrated a few limitations at the User Interface level. Our form of domain restricted query language proved to be not as flexible as we had hoped, especially for novice users. The taxonomy of windows option was also a bit cumbersome to use, simply because the user had to go through several levels before a request could be formulated. Also, we noticed that, specially for requests that might require extended time to be serviced, there is a need to display status information during the different processing stages of a request and to

allow the user to abort a request at any time after it has been delegated to USIA.

Further Work

The above limitations and other proposed features have posed several challenges in the USIA project. Work is already underway in the second phase of development to make progress in two main areas: *ease of use* and *intelligence*. To this end, the following issues and features are being addressed:

- Provide for a two-way *voice interface* for interaction between users and USIA.
- Provide for a full *natural language* processing capability for interface and request delegation purposes.
- Incorporate a capable *user modeling* subsystem which would support modeling of different user types in a multi-domain environment.

Our experience has been challenging but enjoyable. We believe that any progress in this field is bound to have a significant impact on the way people perceive and work with computers.

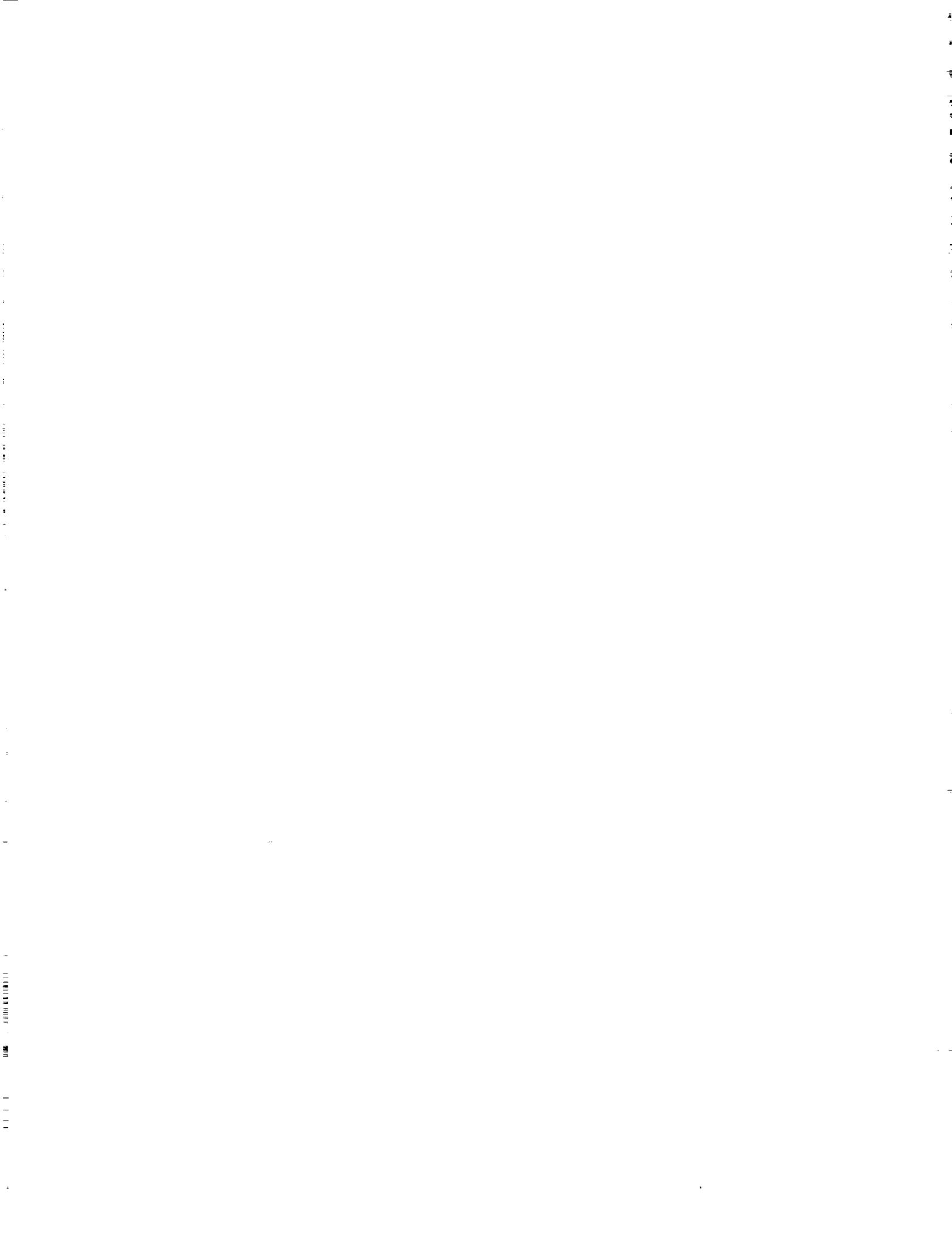
ACKNOWLEDGMENTS

We wish to acknowledge the participation of other members of the USIA project, specially Alexander Glockner, Amit Thukral, Gayathri Rao, Jerome Farley and Mark Flemming. We are thankful for their valuable contribution.

REFERENCES

1. Michael Miley (Agent Technology) MacWeek, Apr. 19, 1993.

2. Yezdi Lashkari, Metral Max, and Maes Pattie. 1994. Collaborative Interface Agents. In *proceedings of the National Conference on Artificial Intelligence*, 444-449, Seattle, Washington: AAAI Press.
3. Yoav Shoham. 1993. Agent-oriented programming. *Artificial Intelligence*, Vol. 60 (1), pp. 51-92, Elsevier.
4. Henry Kautz, Selman Bart, Coen, Michael, Ketchpel Steven, and Ramming Chris. 1994. An experiment in the Design of Software Agents. In *proceedings of the National Conference on Artificial Intelligence*, 438-443, Seattle, Washington: AAAI Press.
5. Norman Donald. 1994. How Might People Interact with Agents. *Communications of the ACM*, Vol. 37, N.7, pp. 68-71.
6. Etzioni Oren, Weld Daniel. 1994. A Softbot-Based Interface to the Internet. *Communications of the ACM*, Vol. 37, N. 7, pp. 72-76.
7. Ted Selker. 1994. "Coach: A Teaching Agent That Learns". *Communications of the ACM*, Vol. 37, N.7, pp. 92-99.
8. Truszkowski, Moore, Odubiyi. 1993. Working paper on "A Multi-Agent-Oriented Approach to Support Satellite Flight Operations Analysis. NASA/Goddard Space Flight Center. Greenbelt, MD.



Building an Adaptive Agent to Monitor and Repair the Electrical Power System of an Orbital Satellite

Gheorghe Tecuci^{1,2} Michael R. Hieb¹ Tomasz Dybala¹

{tecuci, hieb, tdybala}@gmu.edu

¹Department of Computer Science, George Mason University, Fairfax, Virginia, USA

²Center for Artificial Intelligence, Romanian Academy, Bucharest, Romania

Abstract

Over several years we have developed a multistrategy apprenticeship learning methodology for building knowledge-based systems. Recently we have developed and applied our methodology to building intelligent agents. This methodology allows a subject matter expert to build an agent in the same way in which the expert would teach a human apprentice. The expert will give the agent specific examples of problems and solutions, explanations of these solutions, or supervise the agent as it solves new problems. During such interactions, the agent learns general rules and concepts, continuously extending and improving its knowledge base. In this paper we present initial results on applying this methodology to build an intelligent adaptive agent for monitoring and repair of the electrical power system of an orbital satellite, stressing the interaction with the expert during apprenticeship learning.

1. Introduction

Automating the process of building knowledge bases has long been the goal of both Knowledge Acquisition and Machine Learning. The focus of knowledge acquisition has been to improve and partially automate the acquisition of knowledge from human experts by a knowledge engineer. This approach has had limited success, mostly because of the communications problems between the subject matter expert and the knowledge engineer, which requires many iterations before converging to an acceptable knowledge base.

In contrast, machine learning has focused on mostly autonomous algorithms for acquiring and improving the organization of knowledge. However, because of the complexity of this problem, the application of this approach tends to be limited to very simple domains. While knowledge acquisition research has generally avoided using machine learning techniques, relying on the knowledge engineer, machine learning research has generally avoided involving a human expert in the learning loop. We think that neither approach is sufficient, and that the automation of knowledge acquisition should be based on a direct interaction between a human subject matter expert and a learning system (Tecuci, Kedar, and Kodratoff, 1994).

A human expert and a learning system have complementary strengths. Problems that are extremely difficult for one may be easy for the other. For instance, automated learning systems have traditionally had difficulty assigning credit or blame to individual decisions that lead to overall results, but this process is generally easy for a human expert. Also, the "new terms" problem in the field of Machine Learning (i.e. extending the representation language with new terms when these terms cannot represent the concept to be learned), is very difficult for an autonomous learner, but could be quite easy for a human expert (Tecuci and Hieb, 1994). On the other hand, there are many problems that are much more difficult for a human expert than for a learning system as, for instance, the generation of general concepts or rules that account for specific examples, and the

updating of the knowledge base to consistently integrate the learned knowledge.

Over several years we have developed a multistrategy apprenticeship learning methodology for building knowledge-based systems (Tecuci, 1988, 1992; Tecuci and Kodratoff, 1990; Tecuci and Hieb, 1994; Tecuci et al, 1994). Recently we have developed and applied our methodology to building intelligent agents. This methodology allows a subject matter expert to build an agent in the same way in which the expert would teach a human apprentice. The expert will give the agent specific examples of problems and solutions, explanations of these solutions, or supervise the agent as it solves new problems. During such interactions, the agent learns general rules and concepts, continuously extending and improving its knowledge base. This process produces validated knowledge-based agents, because it is based on an expert interacting with, checking and correcting the way the agent solve problems.

Successive versions of this methodology have been implemented in several systems (e.g. DISCIPLE (Tecuci, 1988; Tecuci and Kodratoff, 1990) NeoDISCIPLE, (Tecuci, 1992; Tecuci and Hieb, 1994) and CAPTAIN (Tecuci et al, 1994)), which have been applied to a variety of domains: loudspeaker manufacturing, reactions in inorganic chemistry, high-level robot planning, question-answering in geography and, more recently, military command agents in distributed interactive simulation environments.

In this paper we present initial results on updating and applying this methodology to build an intelligent adaptive agent for monitoring and repair of the electrical power system of an orbital satellite. The system DISCIPLE-OPS, which implements this methodology, provides an integrated framework that facilitates

- 1) building intelligent agents through knowledge elicitation and interactive apprenticeship learning from experts; and
- 2) making these agents adapt and improve during their normal use through autonomous learning.

This paper is organized as follows. Section 2 presents the application domain. Section 3 describes a simulator of the electrical power system to be monitored. Section 4 presents the architecture of the intelligent agent, together with its decision-making and learning methods. Section 5 presents the methodology for building the agent. Finally, section 6 concludes the paper with a discussion of our agent-building approach.

2. An Exemplary Problem

The main objective of the Electrical Power System (EPS) is to provide an Orbital Satellite with a steady supply of electrical power. The EPS is capable of self-preservation in emergencies, but it is not capable of maintaining optimum productivity without outside support. If not controlled, the power production of the EPS will eventually fail, leaving its users unsupported. Therefore, the EPS must be monitored at all times. This function could be fulfilled by an intelligent agent acting as a ground station that monitors telemetry from sensors in the solar powered EPS for anomalous behavior, and generates repairs by forming and uplinking commands to the spacecraft. The agent itself is supervised by a human operator who may correct its behavior. The basic interaction between the spacecraft, the intelligent agent, and the human operator is shown in Figure 1. During such interactions, the agent learns from its own actions and the commands issued by the human operator, gradually acquiring the expertise of the operator until it could operate autonomously.

In the following sections we present a methodology for building the intelligent agent in Figure 1. However, instead of controlling the EPS, the agent will control a simulator of the EPS. This simulator is briefly described in section 3.

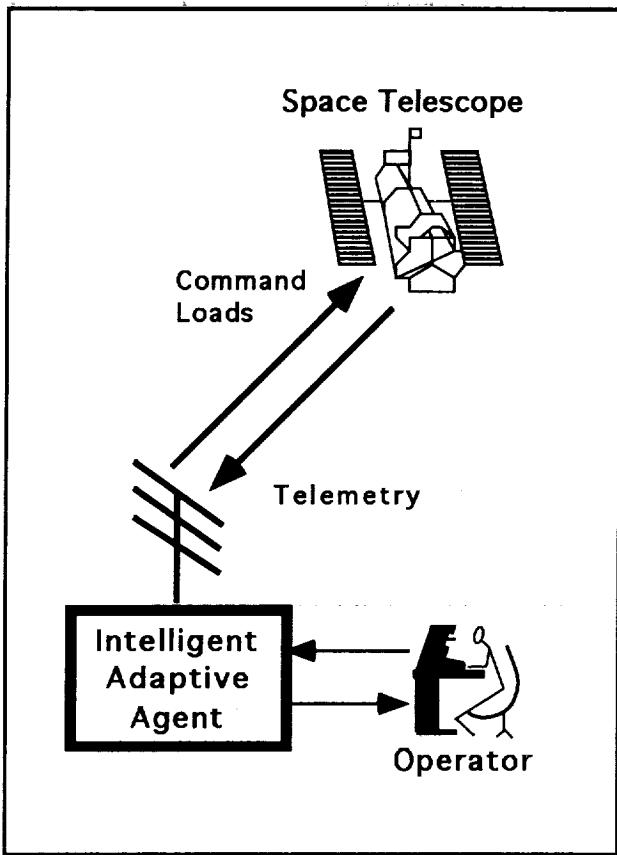


Figure 1. Basic interaction between the spacecraft, the intelligent agent, and the human operator.

3. A Simulator of the Electrical Power System of an Orbital Satellite

A simulator of an Orbital Satellite Electrical Power System has been developed by NASA Goddard Code 522.3 (Silverman et al., 1989; Hieb 1990; Hieb, Silverman & Mezher, 1992). The simulator was not designed to duplicate the EPS of the actual orbital satellite, but is a scaled down version which only simulates selected basic functions and

problems. The goal of the design was to capture the essence of EPS problems and implement them in the simulator. Therefore, this software simulator provides a challenge to the intelligent agent which has close similarities to the real problems encountered at NASA control centers. Figure 2 is a diagram of the EPS simulator. The following components are represented in the simulator.

Solar arrays. There are two solar array panels in the simulator, each with ten solar cells. Power production takes place in the solar arrays. Orientation and cell errors are randomly generated with given certain limits and probabilities. Cell errors are fixed by resetting the appropriate solar array.

The network. The network is a set of power lines equipped with switches and various sensors. The network distributes and directs the power generated by the solar arrays through the system. In the entire network, there are six switches for rerouting current through the system. Switches may cause malfunctions within the EPS. Switch errors are fixed by cycling the specific switch. In this simulation, switches errors are randomly generated. Sensors measure the current at various points on the network. In the entire simulator there are four ammeters and a voltmeter. Network losses are disregarded.

The battery. The battery stores the excess electrical power generated by the solar arrays during the day and then releases it in response to nighttime power requirements.

The bus. The bus represents the load on the EPS. In the simulator the bus power requirements can be adjusted depending on power production or system mission schedule.

Time. A pass, or simulated earth orbit, is always 90 minutes, with 60 minutes of it spent in sunlight.

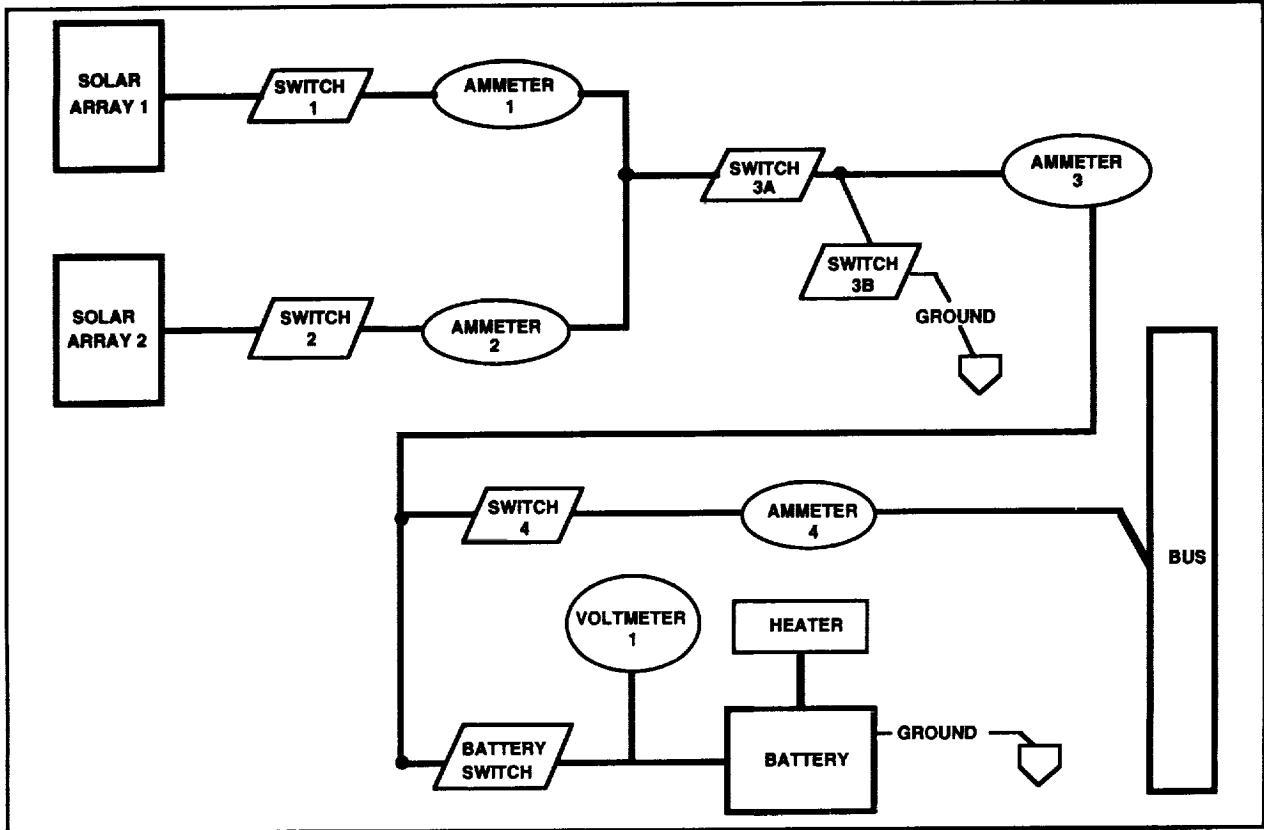


Figure 2. Orbital Satellite Electrical Power System Simulator

4. The Architecture of DISCIPLE-OPS

The intelligent agent, called DISCIPLE-OPS, consists of three main components, the shared knowledge base, the monitoring and repair system, and the multistrategy apprentice learning system, as indicated in Figure 3. The monitoring system uses the shared knowledge base to detect anomalous behaviors of the EPS and to issue repair commands. The learning system extends and corrects the knowledge base as a result of the actions of the monitoring system and the interactions with the human operator.

4.1 The Shared Knowledge Base

The shared knowledge base contains three types of knowledge:

- a hierarchical semantic network representing the electrical power system;
- a set of situation-action rules which detects faults in the EPS and issue repair commands;
- a set of facts representing the current state of the EPS.

A portion of the semantic network from the knowledge base is represented in Figure 4. It consists of a representation of the structure of the EPS, and of the different components of the EPS. This semantic network provides the generalization language for learning.

The knowledge base contains rules of the form:

IF <condition> THEN <action>

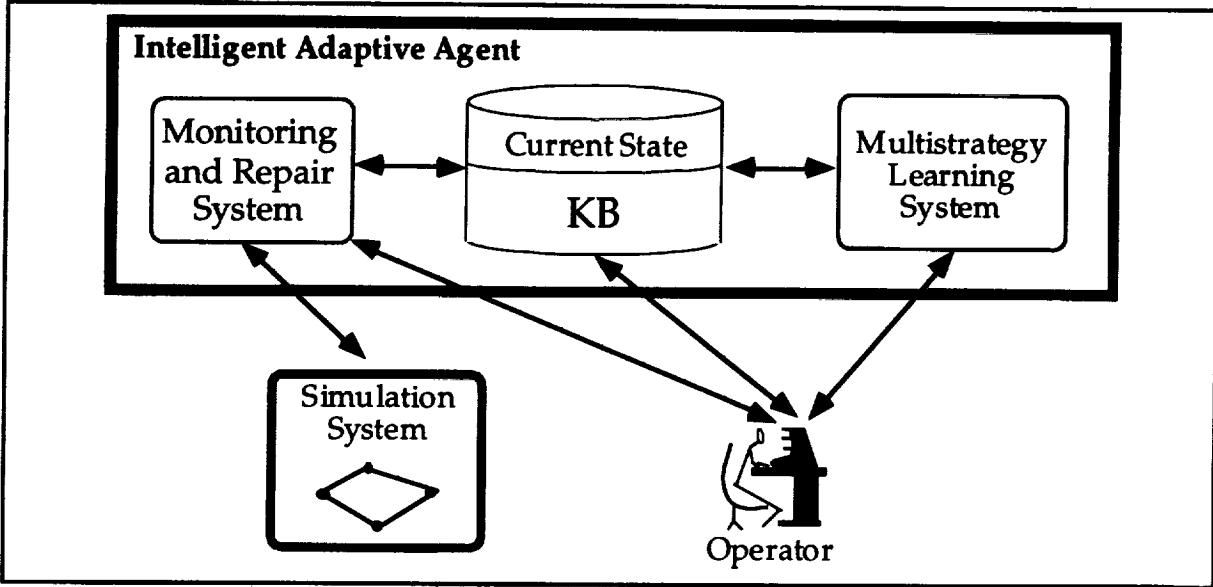


Figure 3. The architecture of the intelligent agent

If the current state of the EPS matches <condition> then the monitoring system will issue the command to perform <action>.

The rules from the knowledge base are learned by the multistrategy apprenticeship learning system from the actions of the human operator. During training many of the rules may not have a single applicability condition, but two conditions, called the plausible upper

bound and the plausible lower bound, as it is shown in Figure 5.

The plausible upper bound is supposed to be more general than the exact (but unknown) condition of the rule, and the plausible lower bound is supposed to be less general than the exact condition. The two bounds define a *plausible version space* [Tecuci, 1992] for the exact condition of the rule.

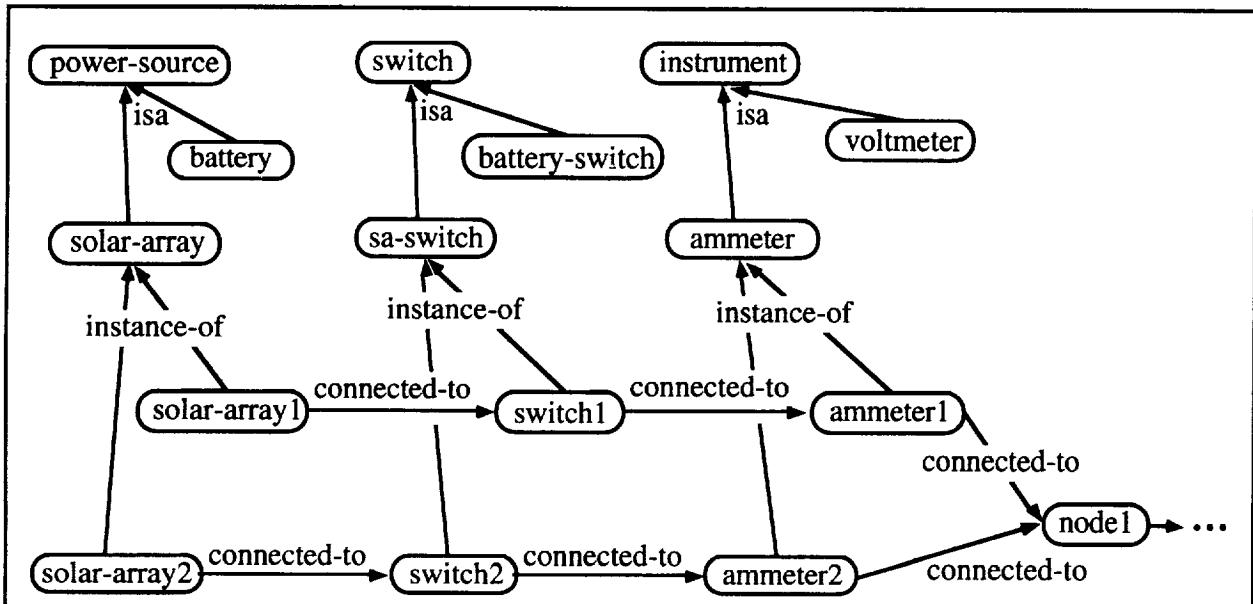


Figure 4. A hierarchical semantic network representing the electrical power system.

IF		
	<i>plausible upper bound</i>	
(ammeter	a (reading low)	; the reading of ammeter 'a' is low
(clock	c (time day))	; during the day
(switch	sw (connected-to a)	; and the switch 'sw'
	(position open))	; between the ammeter 'a'
(power-source	sa (connected-to sw))	; and the power-source 'sa' is open
	<i>plausible lower bound</i>	
(ammeter	a (reading low))	; the reading of ammeter 'a' is low
(clock	c (time day))	; during the day
(sa-switch	sw (connected-to a)	; and the sa-switch 'sw'
	(position open))	; between the ammeter 'a'
(solar-array	sa (connected-to sw))	; and the solar-array 'sa' is open
THEN		
CYCLE	sw	; cycle switch sw

Figure 5. A rule with partially learned conditions.

Each bound is a conjunction of expressions, each expression describing a variable. For instance,

(switch sw (connected-to a) (position closed))

describes 'sw' as being a switch connected to 'a', and being in the 'closed' position. The variable 'a' is described by a different expression from the same bound.

The bounds and the version space are called plausible because they have been initially formed based on an incomplete explanation and its over-generalization (see section 4.3.2).

Also, the learning process takes place in an incomplete representation language that may cause the lower bound to cover some negative examples and the upper bound to fail to cover some positive examples. During learning, the two bounds progressively converge toward the exact applicability condition of the rule. However, due to the incompleteness of the system's knowledge, there is no guarantee that the two bounds will become identical, and therefore equal to the exact applicability condition of the rule. This is not a weakness of the system because it can use the partially learned rules to monitor the EPS system (see

section 4.2), and the rules will be continuously improved.

Finally, the current state of the EPS is represented by the readings of the ammeters and the voltmeter, and the states of the switches (open/closed).

4.2 The Monitoring System

The monitoring system is a situation-action production system, in which each rule recognizes a fault type in the EPS and issues the appropriate corrective action.

If the exact condition of a rule matches the current fault state of the EPS then the action from the right-hand side of the rule is called a *routine repair* of the EPS.

Because many of the system's rules are represented as plausible version spaces, the matching process has to take into account the lower and upper bounds of these spaces.

Let us consider, for instance, the following state of the EPS in which the reading of ammeter1 is low during the day, and switch1 is open (see Figure 2).

The plausible lower bound of the rule in Figure 5 matches the current situation because the following expression is true:

```
(ammeter ammeter1 (reading low))
(clock clock1 (time day))
(sa-switch switch1 (connected-to ammeter1)
           (position open))
(solar-array solar-array1 (connected-to
                     switch1))
```

Because the plausible lower bound of a rule is less general than the exact (but unknown) condition of the rule, the action indicated by the rule (in this case to cycle switch1) is correct. This action will be called a *routine repair* of the EPS.

Let us now consider the case in which the plausible lower bound does not match the current situation. Because this bound is less general than the exact (but unknown) condition of the rule it may still be the case that the exact condition matches the current situation. This can only happen if the plausible upper bound matches the current situation, because this bound is more general than the exact condition. Therefore, if the plausible upper bound of the rule matches the current situation then it is still possible that cycling of switch1 is the appropriate action. This action is an *innovative repair* of the EPS. This repair must be confirmed by the human operator because it is only a plausible solution to the current fault state of the EPS.

Finally, if the plausible upper bound condition of the rule does not match the current situation, then the rule is not applicable.

If no rule applies to the current fault state of the EPS, then the human operator has to indicate a repair action which we call a *creative repair* of the EPS.

One could therefore notice that the plausible version space concept increases system's

flexibility in problem solving, allowing it to perform not only deductive reasoning (based on matching exact or plausible lower bound conditions), but also plausible reasoning (based on matching plausible upper bound conditions).

Therefore, depending on which type of rule condition matches the current situation, the monitoring system distinguishes between three types of repairs of the EPS: *routine repair*, *innovative repair*, and *creative repair*.

4.3 The Multistrategy Apprenticeship Learning System

4.3.1 The learning method

Multistrategy learning is a type of learning which integrates several complementary learning strategies in order to solve more complex learning problems [Michalski and Tecuci, 1994]. Apprenticeship learning is a type of learning from an expert by observing and analyzing its problem solving actions [Mitchell et al., 1985], and is usually based on an interaction with the expert [Tecuci 1988].

DISCIPLE-OPS is both a multistrategy and an apprenticeship learner. A general representation of its learning method is given in Figure 6.

From any creative repair performed by the human operator, DISCIPLE-OPS learns a new situation-action rule which would allow it to make analogous repairs in the future.

First, DISCIPLE-OPS finds an explanation of the creative repair which identifies the important features of the situation. Then, based on this explanation, it defines a plausible version space of a new situation-action rule. This rule is later applied to analogous situations to propose innovative repairs which are accepted or rejected by the human operator.

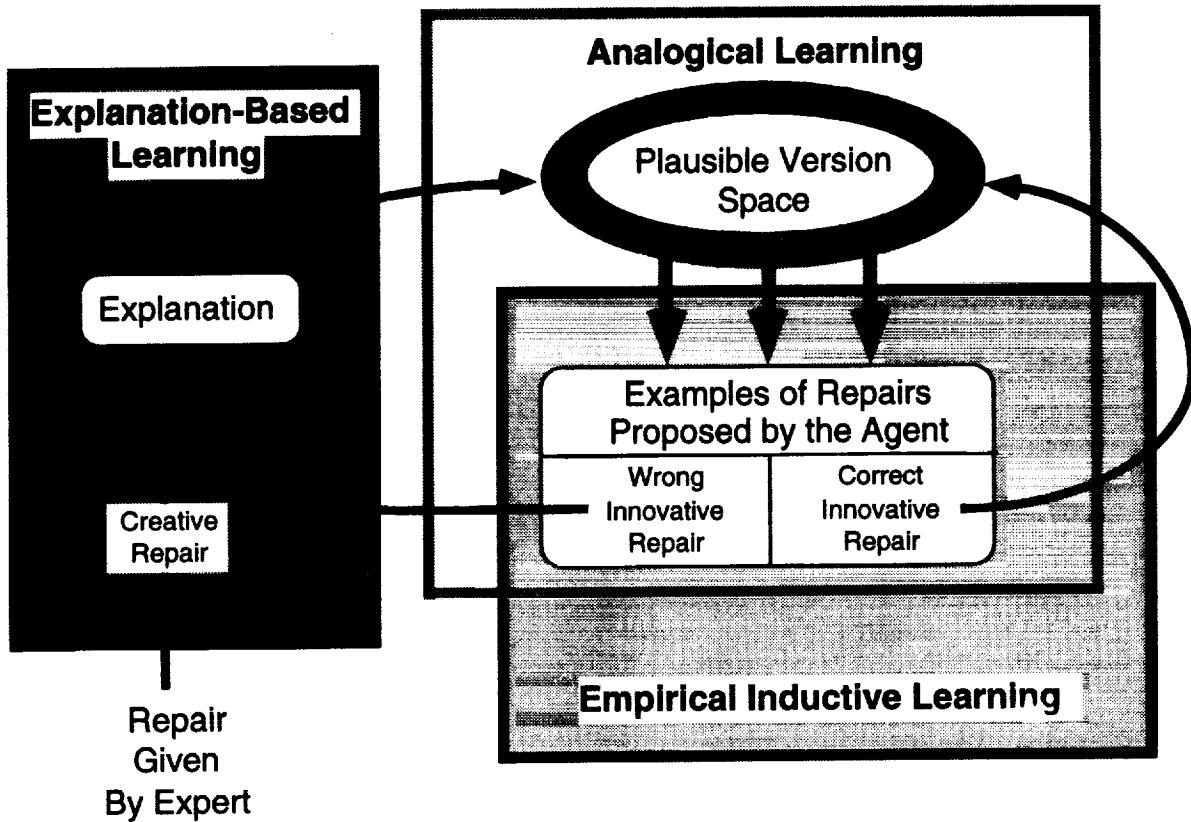


Figure 6. The learning method of DISCIPLE-OPS.

In the case of an innovative repair confirmed by the human operator, the system will generalize the plausible lower bound of the rule so as to cover this repair situation.

In the case of an innovative repair rejected by the human operator, the system will attempt to find an explanation of the failure, and will specialize the plausible upper bound of the rule to no longer cover that situation. In such a situation, the human operator will also have to specify a new creative repair from which the system will learn a new situation-action rule.

The following sections illustrate the different phases of this learning process.

4.3.2 Learning a new rule from a creative repair

Let us consider a state of the EPS for which the human operator proposes the following

creative repair:

CYCLE switch1

First, DISCIPLE-OPS asks the operator to indicate the observations which led to this repair, and receives the following answer:

```
(ammeter1 (reading low))
(clock1   (time day))
(switch1 (position open))
```

Next, DISCIPLE-OPS is trying to find explanations of the fault's cause, in terms of the features and the relationships between the EPS components included in the above observations. It will propose partial pieces of explanations which will have to be accepted or rejected by the operator, as indicated in the following dialog:

Choose the relevant explanations of the current failure:

```
(switch1  (position open)) &
(switch1  (connected-to ammeter1)) &
(ammeter1  (reading low)) ?
yes
(ammeter1  (reading low)) &
(ammeter1  (connected-to node1)) ?
no
(solar-array  (connected-to switch1)) &
(switch1  (position open)) ?
yes
```

The purpose of these explanations is to determine the relevant relationships between the observations and the structure of the network, which will allow the system to recognize similar fault states in the future.

As a result of the above interactions, the following description is identified as characteristic to the current fault state:

```
(ammeter1  (reading low))
(clock1  (time day))
(switch1  (position open))
          (connected-to ammeter1))
(solar-array1  (connected-to switch1))
```

Based on this explanation, DISCIPLE-OPS generates a plausible version space for a new situation-action rule R_i , as indicated in the following.

The plausible lower bound of this rule is just a reformulation of the above explanation, in terms of the variables 'a', 'c', 'sw', and 'sa'. Indeed, these variables can only take the values ammeter1, clock1, switch1, and solar-array1, respectively. Therefore, the lower bound can only match the current fault state (in which it is known that the correct repair is to cycle switch1).

The plausible upper bound is an inductive generalization of the plausible lower bound obtained by turning all the objects into the most general object (called 'something'), turning all the constants to variables, and

keeping the relationships between them.

The purpose of the plausible upper bound is to allow the system to propose innovative repairs in future fault states which are similar to the current one. Examples of these cases are presented in sections 4.3.3 and 4.3.4.

R_i : IF

```
plausible upper bound
(something a (reading x))
(something c (time y))
(something sw (connected-to a)
          (position z))
(something sa (connected-to sw))

plausible lower bound
(ammeter1 a (reading low))
(clock1 c (time day))
(switch1 sw (connected-to a)
          (position open))
(solar-array1 sa (connected-to sw))
THEN
CYCLE      sw
```

4.3.3 Generalizing rules from good innovative repairs

Let us consider a fault state generated by the EPS simulator, characterized by:

```
(ammeter2  (reading low))
(clock1  (time day))
(switch2  (connected-to ammeter2)
          (position open))
(solar-array2  (connected-to switch2))
```

The plausible upper bound of the rule R_i matches this state with the following variable bindings:

```
(a=ammeter2, c=clock1, sw=switch2,
sa=solar-array2, x=low, y=day, z=open)
```

Therefore the monitoring system proposes the following innovative repair (since the variable sw has been instantiated to switch2):

CYCLE switch2

Because this repair is accepted by the operator, the plausible lower bound of the rule R_i is generalized *as little as possible* so as to

cover the current situation and to remain less general than the plausible upper bound. The following generalizations are made, based on the generalization hierarchies from Figure 3:

ammeter1, ammeter2 --> ammeter
switch1, switch2 --> switch
solar-array1, solar-array2 --> solar-array

Consequently, rule R_i becomes:

4.3.4 Specializing rules from bad innovative repairs

Let us now consider a new fault state generated by the EPS simulator, characterized by:

```
(ammeter1    (reading low))
(clock1      (time day))
(switch1     (connected-to ammeter1)
             (position closed))
(solar-array1 (connected-to switch1))
```

The plausible upper bound of the rule R_i matches this state with the following variable bindings:

(a=ammeter1, c=clock1, sw=switch1,
sa=solar-array1, x=low, y=day, z=closed)

Therefore the monitoring system proposes the following innovative repair:

CYCLE switch1

However, this repair is rejected by the operator. In this case, the plausible upper

bound of the rule R_i must be specialized as little as possible so as to no longer cover the current situation and to remain more general than the plausible lower bound.

In this case, the only possible specialization of the upper bound is to specialize the variable 'z' to the constant 'open'. In general, however, there will be many different ways in which the upper bound could be specialized, and the system would need operator's guidance, as illustrated by the following dialogue:

Compare the fault state in which the correct repair is 'cycle switch1'

```
(ammeter1      (reading low))
(clock1        (time day))
(switch1       (connected-to ammeter1)
              (position open))
(solar-array1  (connected-to switch1))
```

with the current fault state in which the correct repair is not 'cycle switch1'

(ammeter1 (reading low))
(clock1 (time day))
(switch1 (connected-to ammeter1)
 (position closed))
(solar-array1 (connected-to switch1))

Which are the relevant differences between the current state and the above one?

(switch1 (position open))

Therefore, rule R_i becomes:

R_i: IF

plausible upper bound

(something	a	(reading x))
(something	c	(time y))
(something	sw	(connected-to a)
		(position open))
(something	sa	(connected-to sw))

plausible lower bound

(ammeter	a	(reading low))
(clock1	c	(time day))
(sa-switch	sw	(connected-to a)
		(position open))
(solar-array	sa	(connected-to sw))

THEN CYCLE sw

The operator also indicates that the correct repair is

RESET solar-array1

Consequently, a new rule, R_j , is learned from the current fault state and its repair, as indicated in section 4.3.2:

R_j : IF

```

plausible upper bound
(something a (reading x))
(something c (time y))
(something sw (connected-to a)
             (position closed))
(something sa (connected-to sw))

plausible lower bound
(ammeter1 a (reading low))
(clock1 c (time day))
(switch1 sw (connected-to a)
             (position closed))
(solar-array1 sa (connected-to sw))

THEN
    RESET sa

```

Rules are continuously improved in this manner, based on positive and negative examples, generated by the EPS simulator. The learning process decreases the distance between the two plausible bounds. The goal

of this process is to make the two bounds identical – at this moment an exact rule is learned. However, because the agent's knowledge is incomplete and partially incorrect, the agent may be unable to learn exact rules and will need to rely on incompletely learned rules, as the one in Figure 5.

4.3.5 Dealing with exceptions

When the agent proposes a routine repair which is rejected by the operator, the corresponding situation-action pair is explicitly associated with the rule, as a covered negative example. Such covered negative examples point to the incompleteness of the agent's knowledge, and are used to guide the elicitation of new concepts and features, by using the knowledge elicitation methods described in (Tecuci & Hieb 1994).

4.4 The monitoring and learning procedure

The procedure in Table 1 summarizes the operation of the intelligent agent.

Table 1. The monitoring and learning procedure.

Monitor:

```

Let S be the current fault state of the EPS simulator
IF the plausible lower bound of a rule  $R_i$  matches S
    THEN issue a routine repair command
ELSE IF the plausible upper bound of a rule  $R_i$  matches S
    THEN issue an innovative repair command
    ELSE ask the operator to issue a creative repair command

```

Learn:

```

IF the operator agrees with the routine repair proposed
    THEN {processing was successful}
    ELSE record the current state as an exception of the rule  $R_i$ 
         ask the operator to issue a creative repair command
IF the operator agrees with the innovative repair proposed
    THEN generalize the plausible lower bound of  $R_i$  to cover the current state of EPS
    ELSE specialize the plausible upper bound of  $R_i$  to uncover the current state of EPS
         ask the operator to issue a creative repair command
IF the operator issued a creative repair command
    THEN learn a new rule for the state S and the repair command issued

```

5 The Methodology for Building a DISCIPLE-OPS Agent

The process of building a DISCIPLE-OPS agent consists of four stages, Knowledge Elicitation, Apprenticeship Learning, Autonomous Learning, and Retraining, as shown in Figure 7. These stages are briefly presented in the following sections.

5.1 Knowledge Elicitation

In the first phase, *Knowledge Elicitation*, the subject matter expert (the human operator) works with a knowledge engineer to define an initial KB which will contain whatever knowledge could be easily expressed by the expert. In the case of the domain considered in this paper, the initial knowledge base consists of a semantic network representing the objects from the EPS (e.g. ammeters, solar-arrays, switches), as well as the structure of the EPS. It will also contain descriptions of the correct states of the EPS, during the day and during the night.

5.2 Apprenticeship Learning

In the second phase, *Apprenticeship Learning*, the agent will learn interactively from the subject matter expert by employing apprenticeship multistrategy learning (Tecuci 1988, 1992, Tecuci et al. 1994), as illustrated in section 4.3. During this phase, the agent's KB is extended and corrected until it becomes complete and correct enough to allow the agent to monitor the EPS autonomously.

5.3 Autonomous Learning

When the agent has been trained with examples of the typical problems it should be able to solve, it enters a third phase,

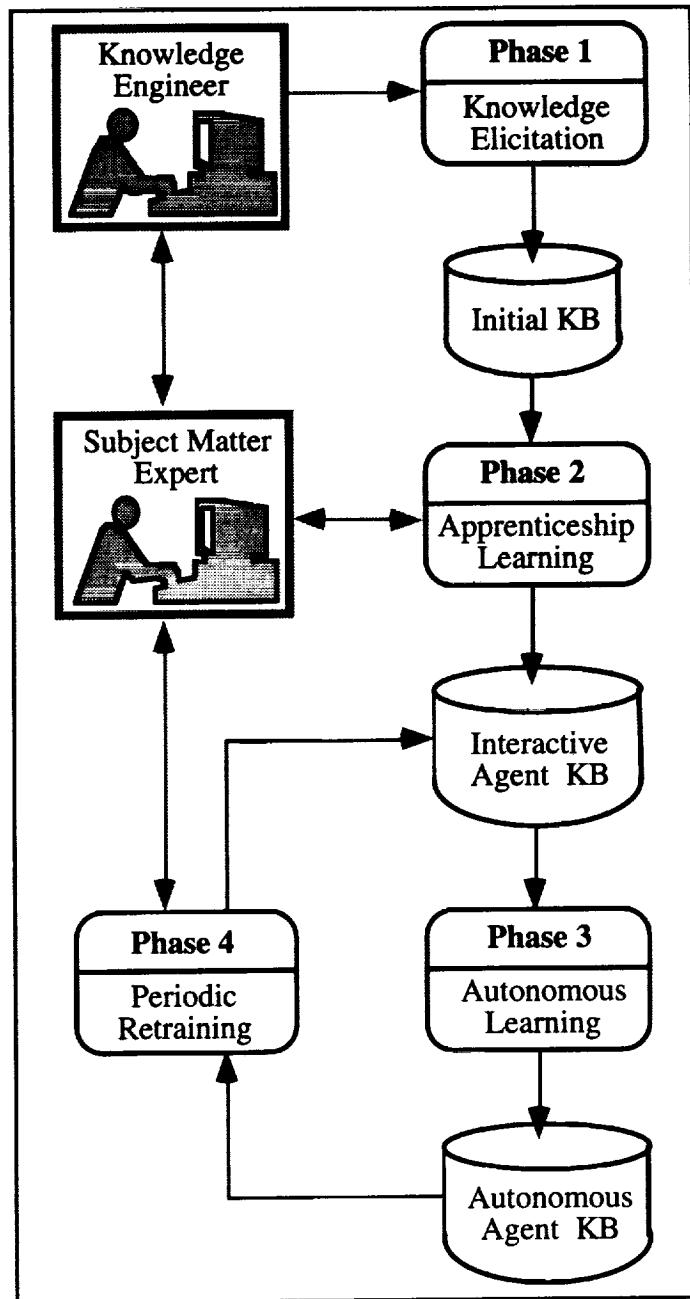


Figure 7. The main stages of building an intelligent adaptive agent.

Autonomous Learning, where it is used to monitor the EPS without the assistance of the subject matter expert. The training received during the Apprenticeship Learning Phase will allow the agent to solve most of the EPS

problems through routine repairs. However, it will also be able to solve unanticipated problems through innovative repairs, and to learn from these experiences, in the same way it learned from the expert. For instance, if the agent issued a successful innovative repair (e.g., applied a rule based on its plausible upper bound condition), it will generalize the lower bound of the rule's condition, to cover the respective situation. If, on the other hand, the agent issued an unsuccessful innovative repair, it will need to specialize the plausible upper bound of the rule. Therefore, the agents developed using this approach will also have the capability of continuously improving themselves during their normal use.

5.4 Retraining

During autonomous learning, the agent accumulates experience and continues to improve its rules. In the same time, it will also accumulate exceptions which correspond to failed routine repairs. After a number of such exceptions have been accumulated, the agent will enter a retraining phase in which it elicits additional knowledge from the operator. Several elicitation procedures which are driven by the goal of eliminating exceptions are described in (Tecuci and Hieb, 1994).

6 Discussion and Future Research

Building intelligent agents is rapidly becoming a major research topic in artificial intelligence (Laird and Rosenbloom 1990; De Raedt et al. 1993; Gordon and Subramanian 1993; Minton 1993; Serge 1993; Van de Velde 1993; Huffman, 1994), due to potential applications of such agents in a variety of domains.

Recently we have been developing a methodology for building intelligent adaptive agents in the framework of our apprenticeship multistrategy learning approach to automated knowledge acquisition (Tecuci 1988; Tecuci and Kodratoff, 1990; Tecuci and Hieb, 1994).

This methodology is currently being implemented in the CAPTAIN system (Hille, Hieb, Tecuci, 1994; Tecuci et al., 1994) which is used to build military command agents for distributed interactive simulations.

In this paper we have presented another implementation of our methodology in the DISCIPLE-OPS system which is used to build operator agents. We have also presented initial results on applying DISCIPLE-OPS to build an intelligent adaptive agent to monitor and repair an electrical power system of an orbital satellite.

Our approach to building intelligent adaptive agents which is illustrated by both CAPTAIN and DISCIPLE-OPS has several advantages. Rather than programming their behaviors in a fixed set of procedures or rules, an expert can train the agent as he would train an apprentice. This will result in the agent acquiring a set of rules that govern its behavior. These rules can later be modified in the same manner as the initial training. Another advantage of this approach is that the expert will verify the agent's behavior during training.

Training efficiency is achieved through the use of simple plausible version spaces (Tecuci, 1992) and a human guided heuristic search of these spaces. The plausible version spaces do not suffer from the limitations of the version spaces introduced by (Mitchell 1978). These limitations are:

- the combinatorial explosion of the number of alternative bounds of a version space (there is only one upper bound and one lower bound in the case of a plausible version space);
- the need to have many training examples for the learning process to converge (significantly fewer examples are needed in the case of our method because the expert's explanations identify the relevant features of the examples);

- the use of an exhaustive search of the version space (as opposed to the heuristic search used with plausible version spaces);
- the inability to learn when the representation language is incomplete (as opposed to our method which can learn partially inconsistent rules).

As illustrated in this paper, the use of plausible version spaces also allows a more flexible type of rule matching. Indeed, the agent may perform a limited type of plausible reasoning to address situations that it has not been specifically trained for.

Although this paper shows the potential application of our approach to building an intelligent adaptive agent for monitoring and repair of the electrical power system of an orbital satellite, much work remains to be done until an effective agent is built. Some of the necessary improvements to be performed are the following:

- defining a better representation of the electrical power system which should also include deeper knowledge of the functioning of the EPS;
- developing the explanation capabilities of the agent, so that it can propose more relevant explanations of a given fault situation. Currently, DISCIPLE-OPS uses only domain-independent heuristics for proposing such explanations. There is therefore a need for identifying domain-dependent heuristics.
- developing a domain-dependent method for generating plausible upper bounds of version spaces from explanations of the initial problem solving episodes. Currently DISCIPLE-OPS uses a domain-independent procedure of turning everything except relationships into variables.

However, the fact that DISCIPLE-OPS has been able to efficiently learn rules relying only on a very general representation of the electrical power system and on domain-independent heuristics, indicates that this approach to agent building may be very successful, if the agent will be provided with a better representation of the domain, as well as more specific heuristics for building plausible version spaces.

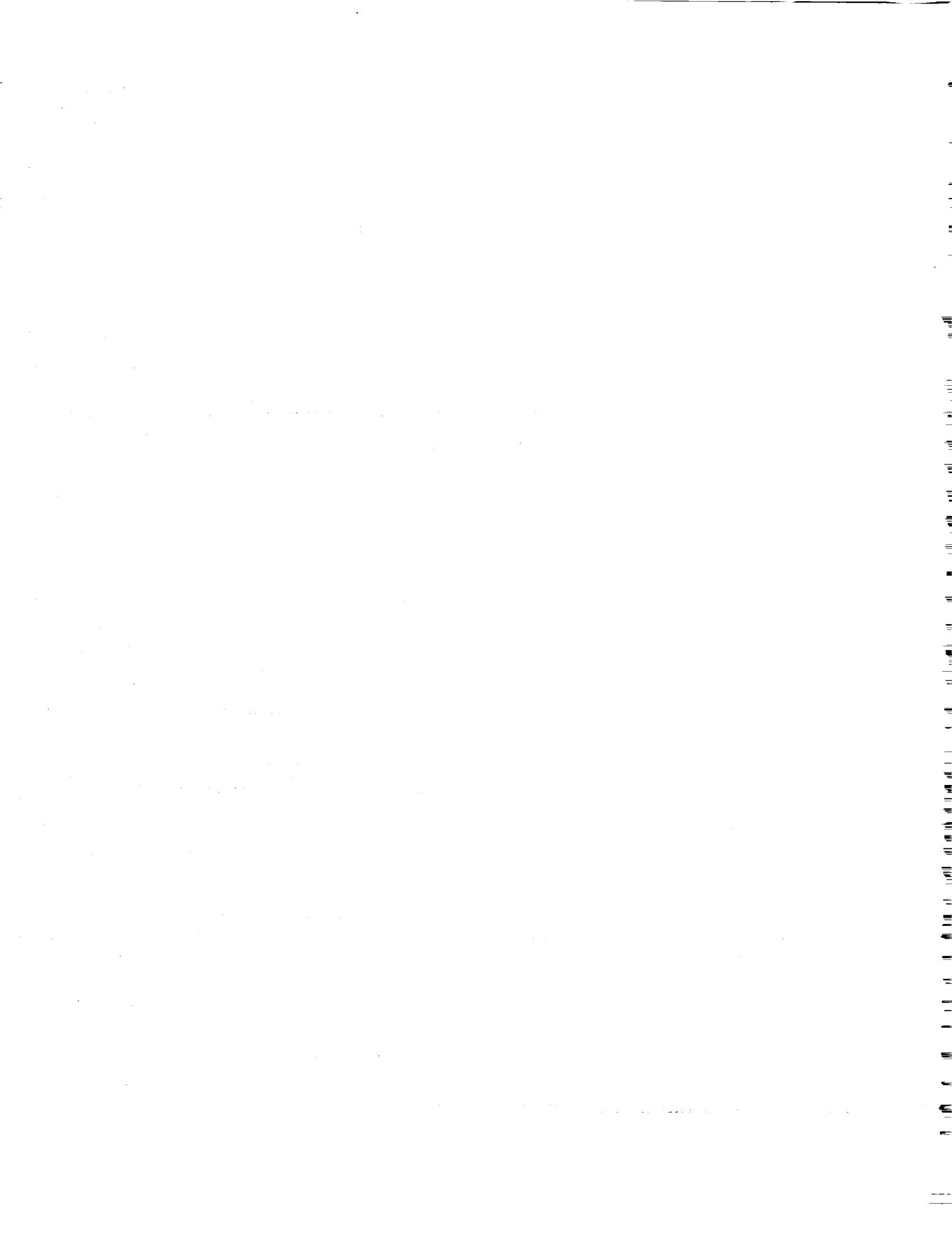
Future research topics also include:

- development of additional forms of consistency driven elicitation, in order to reduce the burden of explanation of the expert;
- development of more flexible methods of instruction that allows the expert to express whatever instruction is desired at any point in the learning process (Huffman, 1994);
- development of methods manipulating and generalizing numbers since the current implementation is based on a translation between numeric parameters and symbolic parameters;
- further development of the problem solving method based on plausible version spaces;
- integration of experience-based learning into the autonomous portion of building the agent.

References

- De Raedt, L., Bleken, E., Coget, V., Ghil, C., Swennen, B. & Bruynooghe, M. (1993). Learning to Survive. *Proceedings of the Second International Workshop on Multistrategy Learning*. Harpers Ferry, West Virginia. 92-106.
- Gordon, D. and Subramanian, D. (1993). A Multistrategy Learning Scheme for Agent Knowledge Acquisition. *Informatica*, 17(4), 331-346.

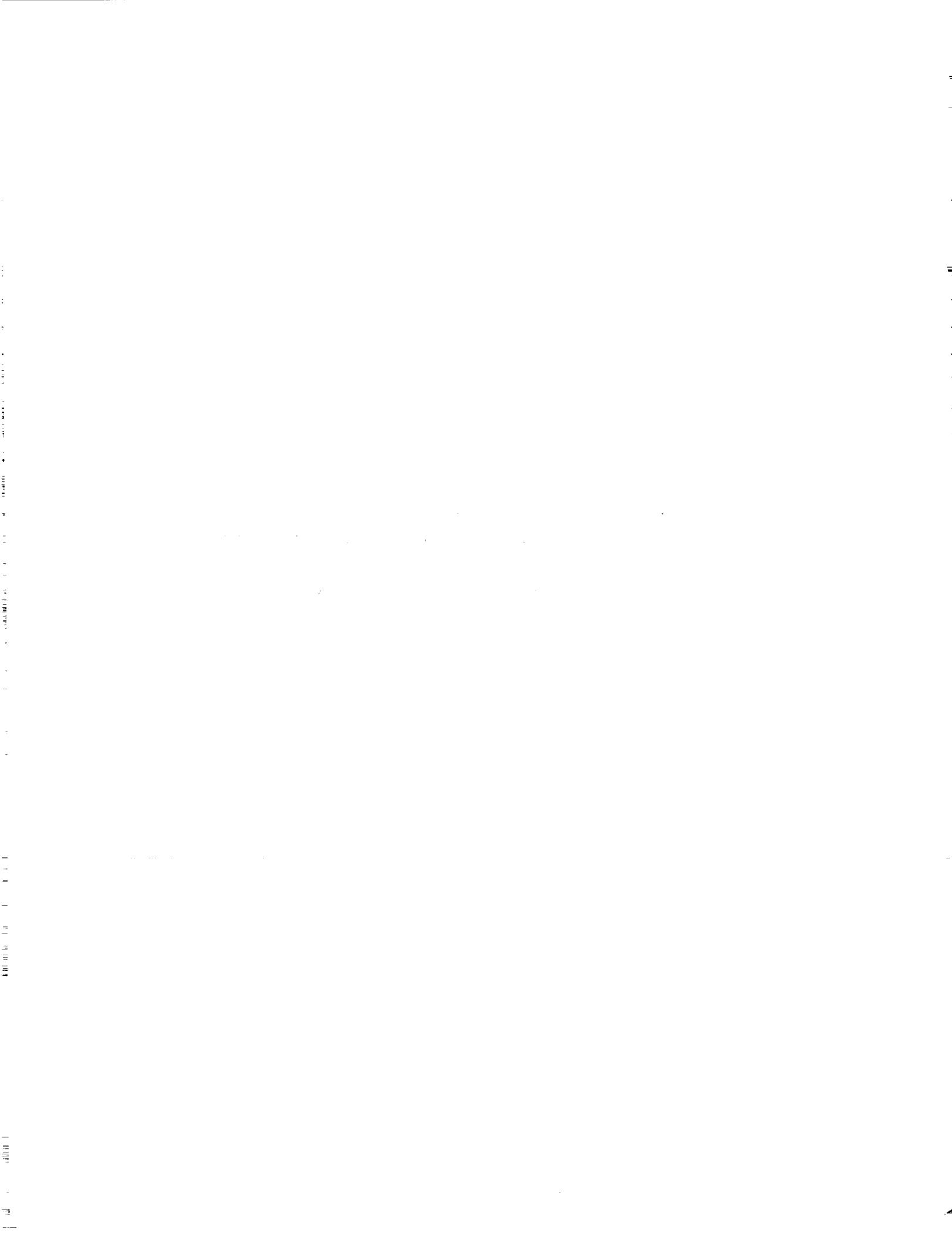
- Hieb, M.R. (1990). Machine Discovery in Large Scale Engineering Systems, *M.S. Thesis*, George Washington University, Washington, DC.
- Hieb, M.R., Silverman, B.G. and Mezher, T. (1992). Rule Acquisition for Dynamic Engineering Environments. *Heuristics – The Journal of Knowledge Engineering*, Winter, 72-82.
- Hille D., Hieb, M.R. & Tecuci, G. (1994). CAPTAIN: Building Agents that Plan and Learn. In *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation.*, May. 411-422.
- Huffman, S.B., (1994). Instructable Autonomous Agents. *PhD Thesis*. Department of Computer Science and Engineering. University of Michigan.
- Laird J.E. & Rosenbloom P.S. (1990). Integrating Execution Planning and Learning in Soar for External Environments, *Proceedings of AAAI-90*. Boston: AAAI/MIT Press, 1022-1029.
- Michalski R.S. and Tecuci G. (Eds). (1994). *Machine Learning: A Multistrategy Approach*, Vol. IV, Morgan Kaufmann, San Mateo.
- Minton, S. (Ed) (1993). *Machine Learning Methods for Planning*, Morgan Kaufmann: San Mateo, CA.
- Mitchell, T.M. (1978) Version Spaces: An Approach to Concept Learning, *PhD Thesis*, Stanford University.
- Mitchell T.M., Mahadevan S. and Steinberg L.I. (1985), LEAP: A Learning Apprentice System for VLSI Design, in *Proceedings of IJCAI-85*, Los Angeles, Morgan Kaufmann, 573-580.
- Serge, A. (1993). Learning how to plan. In W. Van de Velde (Ed.), *Towards Learning Robots*. MIT Press: Cambridge, Mass.
- Silverman, B. G., Hieb, M. R., Yang, H. , Wu., L., Truszkowski, W. and Dominy, R. (1989). Investigation of a Simulator-Trained Machine Discovery System for Knowledge Base Management Purposes. *Proceedings of IJCAI-89 Workshop on Knowledge Discovery in Databases*. Detroit, MI. 327-342.
- Tecuci, G. (1988). DISCIPLE: a theory, methodology, and system for learning expert knowledge. *Ph.D. Thesis*, University of Paris-Sud.
- Tecuci, G. and Kodratoff Y., (1990) Apprenticeship Learning in Imperfect Theory Domains. In Y. Kodratoff, and R.S. Michalski (Eds), *Machine Learning: An Artificial Intelligence Approach*, Volume III, Morgan Kaufmann.
- Tecuci G. (1992). Automating Knowledge Acquisition as Extending, Updating, and Improving a Knowledge Base. *IEEE Transactions on Systems, Man and Cybernetics*. Vol. 22(6), 1444-1460.
- Tecuci, G. (1994). An Inference-Based Framework for Multistrategy Learning, In R.S. Michalski & G. Tecuci (Eds), *Machine Learning: A Multistrategy Approach*, Volume 4, Morgan Kaufmann.
- Tecuci, G. & Hieb, M.R. (1994). Consistency-driven Knowledge Elicitation: Using a Machine Learning-oriented Knowledge Representation to Integrate Learning and Knowledge Elicitation in NeoDISCIPLE. *Knowledge Acquisition Journal*, Vol. 6(1), 23-46.
- Tecuci G. , Kedar S. and Kodratoff, Y., Guest Editors. (1994). *Knowledge Acquisition Journal, Special Issue on the Integration of Machine Learning and Knowledge Acquisition*. 6(2).
- Tecuci G., Hieb M.R., Hille D. and Pullen J.M. (1994). Building Adaptive Autonomous Agents for Adversarial Domains. in *Proceedings of the AAAI Fall Symposium on Planning and Learning*, November.
- Van de Velde, W. (Ed). (1993). *Towards Learning Robots*. MIT Press: Cambridge, Mass.



omit

Planning, Scheduling, and Control

PRECEDING PAGE BLANK NOT FILMED



Limits to Ground Control in Autonomous Spacecraft

Alfred D. M. Wan, Peter J. Braspenning and Gerard A. W. Vreeswijk

56-63

47256
P-15

University of Maastricht, Department of Computer Science
P.O. Box 616
6200 MD Maastricht, The Netherlands

Tel. +31 43 88[2023], [3491], [2021].
e-mail: [wan], [braspenning], [vreeswyk]@cs.rulimburg.nl

Abstract

In this paper the autonomy concept used by ESA and NASA is critically evaluated. Moreover, a more proper ground-control/spacecraft organizational structure is proposed on the basis of a new, more elaborated concept of autonomy. In an extended theoretical discussion its *definitional* properties and functionalities are established. The rather basic property of adaptivity leads to the categorization of behaviour into the modes of *satisfaction* and *avoidance* behaviour. However, the autonomy property with the most profound consequences is *goal-robustness*. The mechanism that implements goal-robustness tests newly generated goals and externally received goals on consistency with high-level goals. If goals appear not to be good instantiations or more acceptable replacements of existing goals, they are rejected. This means that ground-control has to cooperate with the spacecraft instead of (intermittently) commanding it.

1 Introduction

In current spacecraft control engineering two theoretical approaches can be distinguished, viz. [1] equipping spacecraft with autonomy, making them less dependent on ground control and [2] distributing intelligent functions to optimize performance on pre-defined system requirements. Spacecraft autonomy is viewed as a major design goal by leading institutions such as NASA and the European Space Agency (ESA). The main reason for making spacecraft less dependent on ground control, cf. [1], is that total control of the spacecraft is practically unfeasible due to e.g. limited visibility of on-board events. The reason for distributing intelligent functions, cf. [2], is optimization from the point of view of the complete ground-control/spacecraft organization e.g. reduction of operation costs and localization of computational resources, cf. (Grant, 1994; Aarup et al., 1994).

Usually autonomy is defined loosely, which inevitably leads to problems when it is attempted to be used as a design specification cf. (Easter & Staehle, 1984). We critically evaluate the autonomy concept developed by ESA in section 2. From this it will be clear that, before trying to use it, the concept of autonomy needs to be defined first, which is the aim of this paper. The concept of autonomy is developed from contrasting two possible organizational design stances as known in Distributed Artificial Intelligence (DAI), viz. Multi-Agent Systems (MAS) and Distributed Problem Solving (DPS) in section 3, that correspond to the design stances [1] and [2]. In MAS and [1] the emphasis is on autonomy, while in DPS it is on dividing and localizing the functionality of the whole system. It will be pointed out that the functionality of autonomy and the property of independence¹ belong to MAS. Although DPS and MAS may be seen as poles of a continuum, the predominant pole determines both the agent architecture and the organizational possibilities.

In section 4 and 5 we will engage in a full discussion of the origin of autonomy and its functionality. The argument runs as follows: agents that are exposed to uncertain circumstances in which they want to persist have to be adaptive. Systems theory provides an elementary architecture that is maximally adaptive (a feedback system) but has one fundamental inability: it can't change its own goals. Yet, an agent that is based on a feedback architecture can generate or receive new goals, but they have to be instantiations of the unchangeable, high-level goals. In this respect changes to, or generation of goals is restricted, which provides a heuristic warranty to goal approach; this is called *goal robustness*. Goal robustness provides independence from other agents, it will only commit itself to goals that conform with its high-level goals. Independence is thus specified and is a major characteristic of autonomous agents. Finally, we will evaluate what the application in spacecraft architecture of this newly developed concept of autonomy would mean.

2 Spacecraft Autonomy and Automation: a Critical Evaluation

The autonomy concept as developed by ESA (the Standard Generic Approach to Spacecraft Autonomy and Automation; SGASAA, cf. (Pidgeon et al., 1992) was primarily intended to enable spacecraft to continue with their mission, in case of temporary loss of contact with ground control. Any spacecraft that can't be controlled from the ground station and has no means of controlling itself soon perishes. Additional motivations for making spacecraft more independent from ground control, are that due to small communication bandwidths of deep space missions there is little visibility of on-board events and, additionally, long transmission time weaken promptness of the spacecrafts reaction. Also, operation costs would be significantly reduced because there is no need for continuous presence of "marching armies" of ground controllers².

¹Independence is often equated with autonomy, as in (Easter & Staehle, 1984, p. 2-1): "Spacecraft Autonomy: The independence of the man/machine flight system from direct, real-time control by the ground over a specified period of time". In this paper, by independence 'withdrawal from or dismissal of control' is meant. The paper is intended to specify the meaning of independence through defining autonomy.

²Although this quote is taken from a JPL paper (Easter & Staehle, 1984) it also reflects the SGASAA viewpoint rather well.

Basically, the SGASAA concept proposes that the spacecraft should possess a copy of high-level ground-control command sequences or goals (contained in the Master Schedule), so that in the event of a communication failure the spacecraft is able to plan in order to reach the high-level goals. *Independent planning* is done under supervision of an Onboard Management System that, according to the concept, is able to reschedule the Master Schedule, monitors task execution, co-ordinates and controls the various subsystems and payload managers. Planning is hierarchical in the sense that there is a network of plans with at the top the most abstract Long-Term Operations Plan that defines the objectives for an entire mission, and at the bottom Elementary Commands. Also fault diagnosis and recovery should be performed onboard in case a failure coincides with communication loss. There are three modes of operation, viz. [1] routine mode, in which nominal and expected tasks are executed, [2] crisis mode that deals with unexpected events that results in plan failure and [3] check-out mode that checks the proper functioning of the soft- and hardware.

SGASAA has two major drawbacks that raise questions about the alleged autonomy of a spacecraft with SGASAA functionality, viz. [1] the origin of the Master Schedule and [2] the ability of fault diagnosis and recovery. The Master schedule is completely synthesized at the ground station and it consists purely of expandable macro's. The spacecraft only has the abilities to expand the macro's and set parameters, which cannot be called planning. Moreover, ground control can bypass the Onboard Management System and directly command the payloads which would nullify all possible advantages in the case of independent planning, e.g. the adequacy of decisions based on richer knowledge of the actual situation. Concerning the second mentioned drawback, it was known from the outset that only *expected* failures could be catered for. However, failure recovery should, of course, go beyond expected failures.

In spite of the SGASAA aims, the spacecraft remains dependent on the ground station for almost all of its directing functions. In the remaining part of this paper we will develop an alternative concept of autonomy that has a firm theoretical basis and opens up the way to total independent functioning of the spacecraft. We will begin by examining two possible design principles for interacting intelligent systems stemming from Distributed Artificial Intelligence (DAI), viz. Distributed Problem Solving (DPS) and Multi-Agent Systems (MAS).

3 DPS and MAS as Design Principles for Interacting Agents

Distributed AI (DAI) is the field in which systems are designed that have intelligence distributed over a number of distributed nodes or agents. The intelligence consists of knowledge about the problem space (that may or may not be fully specified) and knowledge about problem solving. Applying DAI techniques is useful when the problem under consideration is intrinsically distributed, e.g. geographically when monitoring the movements of vehicles and hypothesizing about their paths, or coordinating the flight movements of aircraft, cf. (Durfee et al., 1987; Durfee et al., 1988; Steeb et al., 1988). DAI systems can generally be designed from two perspectives: Distributed Problem Solving (DPS) or Multi-Agent Systems (MAS). Both systems consist of agents and their organization but DPS takes as its starting

point a particular problem with an adequate organization of distributed nodes, while MAS begins with design specifications of individual agents. (Bond & Gasser, 1988, p. 3) define the two fields as follows:

- DPS considers how the work of solving a particular problem can be divided among a number of modules, or “nodes”, that cooperate at the level of dividing and sharing knowledge about the problem and about the developing solution.
- MAS is concerned with coordinating intelligent behaviour among a collection of (possibly pre-existing) autonomous intelligent “agents”, how they can coordinate their knowledge, goals, skills, and plans jointly to take action or to solve problems.

However, these descriptions, however, don't supply a distinctive criterion for the two fields since there can be many variations of designs that are intermediate. A reason to qualify a system as either DPS or MAS would, in this view, be only a particular stance with which the system is designed. A *top down* design, taking an organizational perspective, would qualify the system as DPS while a *bottom up* design, aimed at designing individual agents, would render a MAS. Figure 1 depicts this view.

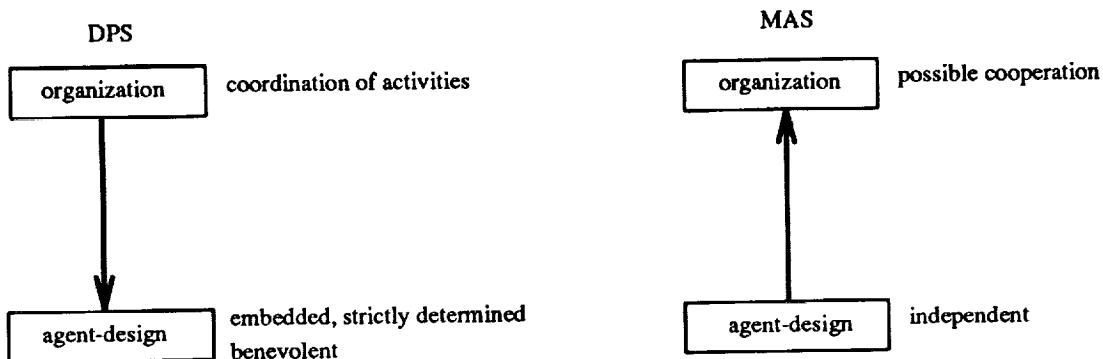


Figure 1: DPS as top-down and MAS as bottom-up designs. The arrows represent typical consequences following from starting-point design restrictions.

(Durfee & Rosenschein, 1994) propose individually different utility functions and means to maximize individual payoff as the fundamental difference between DPS and MAS. This is an underspecification, however, because agents can be designed in such a way that utility maximization of the individuals contributes to the higher order, organizational goal. Durfee and Rosenschein therefore consider MAS and DPS properties in more detail to be more specific about the *self-interestedness* criterion for MAS agents.

Another way to look at the difference is by considering DPS as a subset of MAS through a certain number of extra assumptions that hold for DPS, viz. [1] the *benevolence assumption*, i.e. agents are willing to help each other whenever possible, [2] the *common goal assumption*, i.e. DPS agents all have the same goal which may be explicitly represented but may also be embedded in the organization and possible roles agents can assume under certain

circumstances, [3] the centralized designer assumption, i.e. the designer controls all aspects of agent behaviour in a fully specified environment. To summarize: in DPS the agent design is completely dependent upon the higher order goal and behaviour is completely determined by organizational choices meant to solve a particular problem.

There are a few problems with these criteria. First, as Durfee & Rosenschein observe, even with the mentioned assumptions, DPS systems do not necessarily function optimal due to non-optimal local decisions made by the individual nodes. Also, the extent to which the goals should differ to categorize them as MAS agents is unclear. Finally, it is possible to equip agents with a payoff function that instantiates a high-order goal which makes it difficult to decide whether they are MAS or DPS agents. In fact, this was the reason to view DPS and MAS designs as spanning a continuum in which both designs may have different starting points. Table 1 contains a summary of the contrasting properties of DPS and MAS.

Table 1: Contrasting DPS and MAS properties

DPS	MAS
agent design depends on organizational choices	pre-existing / pre-formed agents
benevolent	self-interested
common goals/utility function	individual goals/utility function
fully specified environment	unspecified / partly specified environment
organizational coordination of results and / or tasks	societal cooperation on basis of joint plan formation
global success criteria	situation assessment based on goal states
domain-specific problem solving	individual problem solving / self-maintenance

Hence, neither the individual utility function, nor the restrictions on MAS that define DPS, provide a decisive criterion. The problem is that although the possession of individual utility functions seems a good candidate, it is difficult to determine whether they are dependent on the social goal. In fact, goals in artificial agents are always dependent on the goals of the designer³.

Thus, we argue that the only way to be certain that the agents are *self-interested* and that their goal structure doesn't depend on the designer's, is that the agent's goal structure has evolved from scratch in a real environment. This is the case only for living organisms. Hence, *autonomous agents cannot be designed* but rather have evolved by themselves. Still, to be practical, we can maintain a notion of *quasi-autonomy*. An agent is quasi-autonomous if it mimics the functionality of autonomous agents. Autonomous agents have to preserve themselves, which is the first, and necessary, requirement for self-interestedness. Self-preservation, in turn, requires adaptability.

³We can be quite fussy about this point and argue that even if agents have random goals, they are still dependent on the designer's goals, because he or she has had reasons (motives) to design the agents' goal structure in that particular way.

It will be clear by now, that SGASAA functionality doesn't come near quasi-autonomous functionality. SGASAA is much closer to the DPS pole, because of the limitless way ground control can influence the goals of the spacecraft. Nevertheless, taking the fact into account that spacecraft indeed face uncertain circumstances and that they should preserve themselves (which is one of the motivations for SGASAA), its design actually *should* be closer to our concept of (quasi-)autonomy. In the next section we will again consider the origin of the goals of adaptive systems and further examine adaptive functionality from a theoretical perspective to specify what autonomous agents should be capable of.

4 Autonomy as Resulting from Constraints on Adaptation

In this section we will give an explanation of the autonomy feature of independency that in our view arises from the property of adaptivity. The first observation, drawn from traditional systems theory, will be that maximally adaptive systems are goal-directed, i.e. they try to lift the discrepancy between measurements from situations and an internally represented goal-state using feedback to guide action. Secondly we will address the origin of the goals in adaptive systems. Since certain goals can't be changed agent-internally, we also have to look at the goal development from a intra-specimen perspective, which will be called the *fylogenesis* of goals.

Finally, we will look at a possible architecture that takes elementary adaptive units as its building blocks to comprise a more general adaptive system that besides the already facilitated functionality of goal achievement, which we shall call *satisfaction behaviour*, also displays goal patching or avoidance behaviour when the threat that a particular goal will not be achieved becomes too big.

In this section, we primarily establish that adaptive systems contain a core of stable goals, i.e. a core that can't be influenced. In the next section we will look at what that means for a system that is capable of generating goals endogenously or capable of receiving external goals. Figure 2 summarizes what issues will be considered in what order in this and the next section.

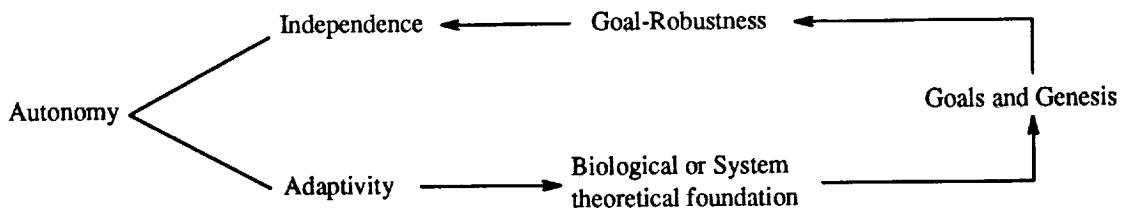


Figure 2: The two parts of autonomy and their relationship. Independence is justified by the MAS property of adaptiveness to uncertain circumstances.

4.1 First Ramifications of Adaptivity: a Systems Theoretic Approach

Traditionally, system theory, cf. (Glisson, 1985), roughly divides (linearized) systems into three categories, viz. [1] I/O systems [2] systems with a state representation or a state vector [3] feedback systems. I/O systems doesn't have a 'memory' and their output is a (linear) transformation of the input. Systems with a state representation have an output function which is dependent on the previous state and possibly a direct component from the input. In feedback systems, output is directed back through a function that contrasts output with desired output. A block-diagram of an output-feedback system can be found in figure 3, cf. (Owens, 1978).

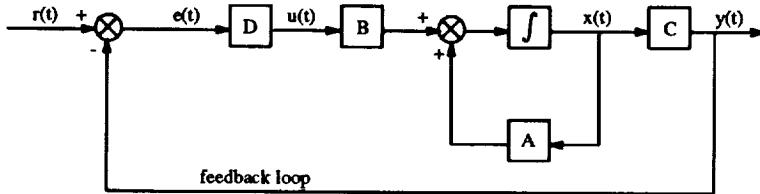


Figure 3: A constant output feedback-control system.

The system equations are as follows:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), \quad x(t) \in R^n \\ y(t) &= Cx(t), \quad y(t) \in R^m, \quad u(t) \in R^l\end{aligned}$$

The state vector is denoted by x , together with corresponding transformation matrix A it comprises a 'memory'; u is the input vector with transformation B ; y the output vector that depends on the state through transformation C . A $m \times 1$ vector of demanded outputs, $r(t)$ can be constructed, to result in an error vector $e(t) = r(t) - y(t)$ that is fed back into the system, resulting in the input $u(t) = De(t)$ where D is a constant $l \times m$ matrix.

An output feedback control (OFC) system is a straightforward extension of the two simpler systems. In addition to the I/O model, 'memory' is added which makes iterated action possible, and in addition to the state-vector system feedback is added, which makes it possible to compare output with desired output.

Comparing output with demanded output is a test for performance, or a test to what extent the situation converges towards the desired state. Demanded output can thus be seen as a goal and feedback gives an indication of how closely the system has neared the goal-state⁴. The limitations of the OFC system are that only information is processed that a priori was established as goal-relevant.

According to a classification of (Cariani, 1991, p. 786), there are two types of systems below

⁴This system-theoretic classification can very well be mapped on the classification of (Genesereth & Nilsson, 1987). They distinguish [1] tropistic agents (I/O systems) [2] hysteretic agents (state vector systems) and [3] knowledge-level systems (OFC systems).

the adaptive one that don't dispose of the capacity to learn or use a feedback design, viz. [1] fixed computational and [2] fixed-robotic. They do have the capacity to respectively execute pre-specified rules and execute fixed percept-action combinations but not to optimize their percept-action coordination. He calls the OFC system the adaptive device type and there is another more general adaptive system, viz. the general evolutionary device type. Cariani's classification can be found in table 2. In the next subsection we will focus on the difference between the adaptive and the general evolutionary system in order to establish the maximally feasible adaptive system and its properties.

Table 2: Device types according to Cariani, a representative of A-Life

Device Type	Capacities	Limitations
fixed computational	reliable execution of pre-specified rules	limited to pre-specified rules and states
fixed robotic	reliable execution of fixed percept-action combinations	no feedback or learning from the environment
adaptive	performance-dependent optimization of percept-action coordination	limited to percept & action categories fixed by the sensors & effectors
general evolutionary	creation of new percept & action categories: performance-dependent optimization within these categories	time to construct & test new sensors & effectors may be very long

4.2 Passing Fitness Criteria through the Genome

As we have noted in section 3 the genesis of the goals in adaptive systems plays a key role in the notion of autonomy. To be genuinely autonomous, the goals of an agent should originate from the objective of self-preservation in an uncertain environment. The issue we will now consider concerns the process of genesis, i.e. how goals can evolve and especially the question whether individual agents are capable of changing or generating all of their goals themselves (i.e. *intra-specimen*) or that change happens *inter-specimen*.

According to Cariani there exists an adaptive device that is more general than an OFC, viz. the evolutionary device type. This device is capable of the development of new sensors, establishing new computations on new sensory primitives (what he calls *epistemically autonomous*, we will not consider this property further) and constructing their own performance-measuring apparatuses (this is what he calls *motivationally autonomous*, Ibid. p. 789). Motivationally autonomous agents change their evaluative criteria (what we have called norms or goals) themselves.

OFC systems are directed towards the norm, they evaluate output using the norm. If they would be able to change their norm, there would be no guiding criterion because the norm itself has that function. This means that systems capable of changing the norm must do so arbitrarily. For systems that must maintain themselves changing the goal that they want to achieve arbitrarily involves a high risk. Goals that are generated arbitrarily may direct the system to self-destruction. Without a stable prior goal, there is no way new goals can be tested on adversity or beneficence. It could even be argued that the general evolutionary

device type can't be classified as an adaptive system because there is no demanded output, after all, if something is demanded then a random mutation of it is not necessarily demanded.

If we look at the evolution of natural agents, experiments that randomly change the architecture of organisms take place but *inter-specimen* rather than *intra-specimen*. Mutations take place from one generation on to the next and the success of this experiment is determined by fitness criteria⁵. If a specimen matches the fitness criteria well enough, the changes are passed through the genome and remain stable in the next specimen. Comparing possible architectures in terms of systems theory leads to the conclusion that OFC systems have an advantage because they are goal-directed but only if their norm is a proper representation of environmental survival conditions, i.e. if the norm has developed under evolutionary conditions. We call the evolutionary development of goals the *fylogenesis* of the goals.

Changing evaluative criteria arbitrarily is unpermissible for individual specimens because it leaves them without any success criteria of their action which would nullify the advantages of feedback. From this we conclude that a more general adaptive device through flexibility of the totality of goals, is not feasible. However, there is an extension of the adaptive device that consists of layers of OFC systems and has important, indispensable functionality. We will turn to this now.

4.3 Reflectiveness in Task Networks of OFC systems

Although Cariani's evolutionary device type doesn't provide a more general adaptive system than the OFC, the adaptive functionality of an OFC can be extended. Briefly this can be achieved by recursively linking the OFC systems into a *task network* so that there is a hierarchy with at the top level OFC systems representing the fylogenetic goals (we will call these *primitive goals*) and at the bottom level OFC systems that perform action primitives and intermediate subgoals⁶.

In comparison to classical planners, cf. (Charniak & McDermott, 1985, ch. 9), task networks consisting of OFC systems have an important advantage over traditional planning operators. In traditional planning theory monitoring the execution of a plan is identified as one of the most intricate problems (Ibid. p. 489, 524). The decision when to abandon a subgoal (i.e. assessment of the situation to establish the rate of convergence to the goal state), is of prime importance in plan-execution monitoring. This decision can be based on the rate of goal-state convergence. However, a single OFC system is not able to change its action pattern; in fact, it can only amplify its attempts. Hence, if convergence is too slow or if the situation diverges from the goal-state, there should be a possibility to change the action pattern altogether.

The creation of a task network with OFC systems is straightforward because each OFC has

⁵Fitness is defined in (Koza, 1992, p. 94) as the probability that an individual survives to the age of reproduction and reproduces. In Artificial Life there are usually other methods of measuring fitness than reproduction, e.g. in a population of artificial ants it can be the number of food parcels eaten.

⁶We will not elaborate much on the linking mechanism because of space limitations. An elaborate discussion of this can be found in (Nilsson, 1994).

a description of its goal state in r . If π is the overall goal and there exists a schedule of $\{r_1 \cup r_2 \cup \dots \cup r_n\} \supseteq \pi$, a task network for π exists. Execution of any current subgoal can be monitored by examining if the output converges to its desired state:

$(r_p - y_p(t)) - (r_p - y_p(t+1)) \leq c$. Using an abstract matching operator M , then if there is an alternative goal r_q such that $r_q M r_p$ and $x_p(t) \neq x_q(t)$, then an alternative task network can be reassembled on the failure of a subgoal. We will call this the 'reflective' property of the system.

In reflective systems two modes of behaviour can be distinguished: *satisfaction behaviour* in which a task network is synthesized after which all the subgoals are attempted to be satisfied and *avoidance behaviour* in which replanning is initiated to patch up failing (sub)goals. Sole OFC systems are not able to find alternatives for their attempts to satisfy their goals, while reflective systems are (in principle) able to apply alternatives.

With the property of reflectiveness we have completed our specifications for a system that has maximal adaptivity. Before we continue to examine its properties, we will make one last remark about the epistemological status of the ontogenetic goals. In the previous section we argued that a system that has no fixed evaluation criteria is undirected and that the criteria are fixed by environmental constraints passed on through the genome. We now have a network of OFC systems that interact through their goal-state descriptions (the respective r 's). Because the OFC systems are hierarchically organized, higher level goals can act as supervisors of lower level goals and change their goal-state depending on performance which creates possibilities for percept-action coordination that Cariani only granted the general evolutionary device type⁷.

5 Goal-Robustness, Independence and Autonomy

In the previous section the two main functional modes of goal-directedness, i.e. satisfaction and avoidance behaviour were discussed, and the canonical position of the goals was established through examining the adaptive design. The canonicity of the goals has another implication, viz. the property of goal-robustness. If a system has the property of goal-robustness all of its goals are instantiations of a set of irreducible or primitive goals of which the existence was shown in the previous section. The issue in this section is the functionality of the system that can generate endogenous or receive external goals.

Section 5.1 demonstrates the connection between the autonomous feature of independence as proposed by (Castelfranchi, 1994) and the property of goal-robustness. Independence derives from a mechanism that implements goal-robustness, i.e. only those goals are assimilated and scheduled that are consistent with the primitive goals. Finally, the criteria

⁷His claim that a system is only truly emergent if a system places itself outside the observational frame of the designer is, according to our opinion, mistaken. (Rosen, 1986) has made a case for the informational equivalency of behavioural determining factors, i.e. genome and state, that shows the fundamental impossibility of reducing observations to one of these factors as an explanation. In fact, this is a methodological problem of underdeterminedness of observations through which it is in principle not possible to make a distinction between the adaptive - and the general evolutionary device type.

for assimilating directives from other agents are formalized which completes the features necessary for autonomy. Figure 4 schematically shows the canonical position of the goals.

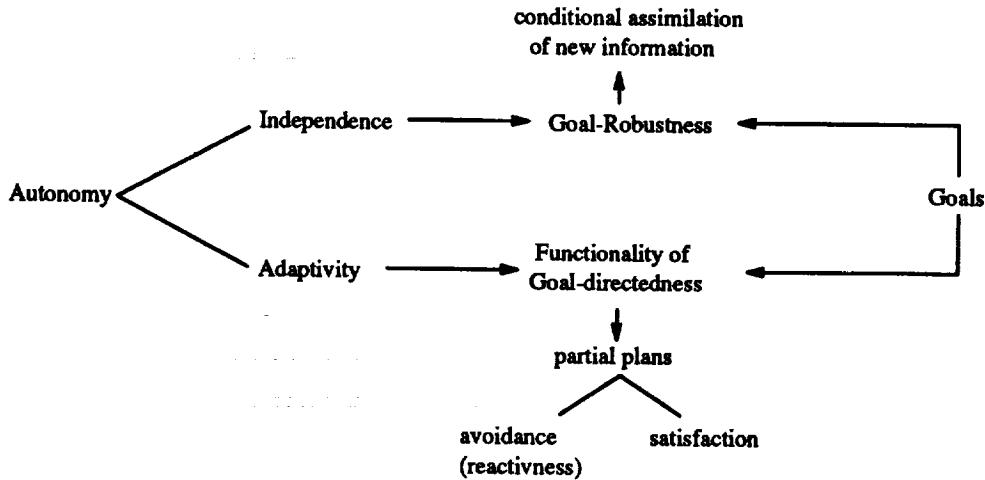


Figure 4: The two parts of autonomy and their relation. Independence is justified by the MAS property of adaptiveness to uncertain circumstances.

5.1 Criteria for Autonomy

According to Castelfranchi there are two kinds of autonomy, viz. [1] autonomy from physical context (from environment) and [2] autonomy from social context (from other agents). He refers to [1] as the "Descartes Problem": "Which Agent Architecture guarantees that the Agent is neither completely determined by stimuli (stimulus dependent), nor completely unreactive to environmental changes?" (p. 52).

We interpret [1] as the question of how action can originate from the agent, i.e. endogenously rather than completely from the current situation. The embedded feedback systems architecture, discussed in the previous sections, provides a solution. All the goals in the planning system are instantiations of the phylogenetic goals, and therefore provide an explanation of behaviour that doesn't originate from the current situation. Thus, the task network combines situational appropriateness and conformity to the phylogenetic goals⁸. The execution of the task network is constantly being monitored, adjusting when necessary and even abandoning the current action pattern if it appears to be inadequate. Execution monitoring and adjustment provides the required reactivity. With the ability to pursue endogenous goals, an agent could be said to be independent from its environment. Actually, the property that is intuitively closer to independence is independence from other agents, which is Castelfranchi's second criterion.

Autonomy from social context means that an agent will not unconditionally follow goals others propose to it. This follows naturally from the fact that primitive goals are stable and

⁸Explanation of behaviour of autonomous agents is diachronical rather than synchronical as it is in I/O systems.

that the agent's goal-state representations should be consistent, i.e. not have contradictory goals, we will call this the property of *goal-robustness*. Providing the agent with goal consistency is a distinct problem that requires various goals to be contrasted along a consistency measure. It is clear that a consistency measure should be a function of the subjected goal and the primitive goals. In the next subsection we will look at what this function might look like.

5.2 Checking New Goals on Consistency

The question of how the goal state of individual agents changes when agents try to influence each other's goals explicitly, i.e. through communication of directives, is addressed by (Werner, 1989). He contends, as we do, that in real life situations complete plans cannot be communicated (*Ibid.* p. 7) and that goal states can't be changed by others unconditionally (*Ibid.* p. 17). However, except for a few well-defined cases in which the organizational structure determines the conditions under which new goals are assimilated (so called *roles*), he doesn't define a function that tests received goals on compatibility with agent-dependent utility functions. We will make a first attempt in order to specify the property of goal-robustness that was introduced in the previous section.

Task networks are usually assembled by searching for task operators that reduce the difference between the current - and the goal state (cf (Charniak & McDermott, 1985, ch. 9)). The result is a *goal conjunction* π that can be matched against the current situation represented in I . In π there may be a number of variables, either to be bound to other operators or to primitives in the situation representation, we will denote this as follows: $\pi\theta$ in which θ represents the set of free variables⁹. On assembly of the plan, the set of differences can be reduced by replacing goal conjuncts by conjuncts that have greater detail and therefore a better match with the situation, until the planner has found a *maximal match substitution*. In effect this is a reduction of the number of free variables to a substitution such that for all other substitutions θ' , $|\pi\theta - I| \leq |\pi\theta' - I|$.

In a task network the leaves of the planning tree are matched in this way to the situations. However, we do not only want to know how well a task network fits a particular situation but also how well a particular tasks network instantiates the primitive goals, which is ultimately where the agent is directed at. Any subgoal can be tested on fit with a higher-level goal from which a utility value is produced which is maximal when the match is perfect, i.e. when there are no free variables. On perfect match, a task network will completely be executable and it will realize a high-level goal. Analogous to the matching of a task network to a particular situation, a particular instantiation $\pi'\theta$ can be matched to a higher-level goal π . Analogous to maximal fit, maximal utility with respect to goal π is defined as follows:

$$\forall\theta' \exists\theta \{|\pi'\theta - \pi| \leq |\pi'\theta' - \pi|\} \quad (1)$$

For a perfect plan the following holds: $\pi'\theta \subseteq \pi \subseteq I$. Overall utility of the instantiation is

⁹Our notation is a mixture of Charniak & McDermotts, of Werners and our own.

defined as follows:

$$E(\pi, \pi', \theta) = \frac{1}{|\pi' \theta - \pi|} \cdot \lambda_\pi \quad (2)$$

In which λ_π is an overall utility value of π or a constant if π is a direct instantiation of a primitive goal. Hence there exists a set of fixed utilities, Φ that is a subset of the total set of utilities and which is indexed to the set of primitive goals Ψ : $\Psi \rightarrow \Phi$; $\Psi \subseteq \Pi$ and $\Phi \subseteq \Lambda$.

The evaluation function enables the planner to decide which goal and instantiation to choose. The utility of a particular goal depends recursively on the match with a primitive goal, hence goals that instantiate a highly rated primitive goal well are preferred above goals that are either poorer instantiations or linked to lower rated primitive goals. In an environment where the planning agent is liable to influence, the evaluation function provides a strong criterion for assimilation of a communicated goal. There are two possible situations: either the received goal is an instantiation of a priorly uninstantiated goal, or it is a replacement of an already existing goal. In the first case the criterion for assimilation is that total utility must increase: $\sum_n E(\pi_i, \pi'_i, \theta) > \sum_n E(\pi_i, (\pi'_i + \pi'_q), \theta)$ in which n equals the total number of goals. In the second case the criterion is that if the received goal can be instantiated so that it has a higher evaluation value than an already present goal, it will be assimilated into a task network and executed, otherwise it will be rejected. Formally, the criterion for accepting the new goal π'_2 at the cost of goal π'_1 is: $E(\pi, \pi'_1, \theta) \geq E(\pi, \pi'_2, \theta)$.

6 Conclusions

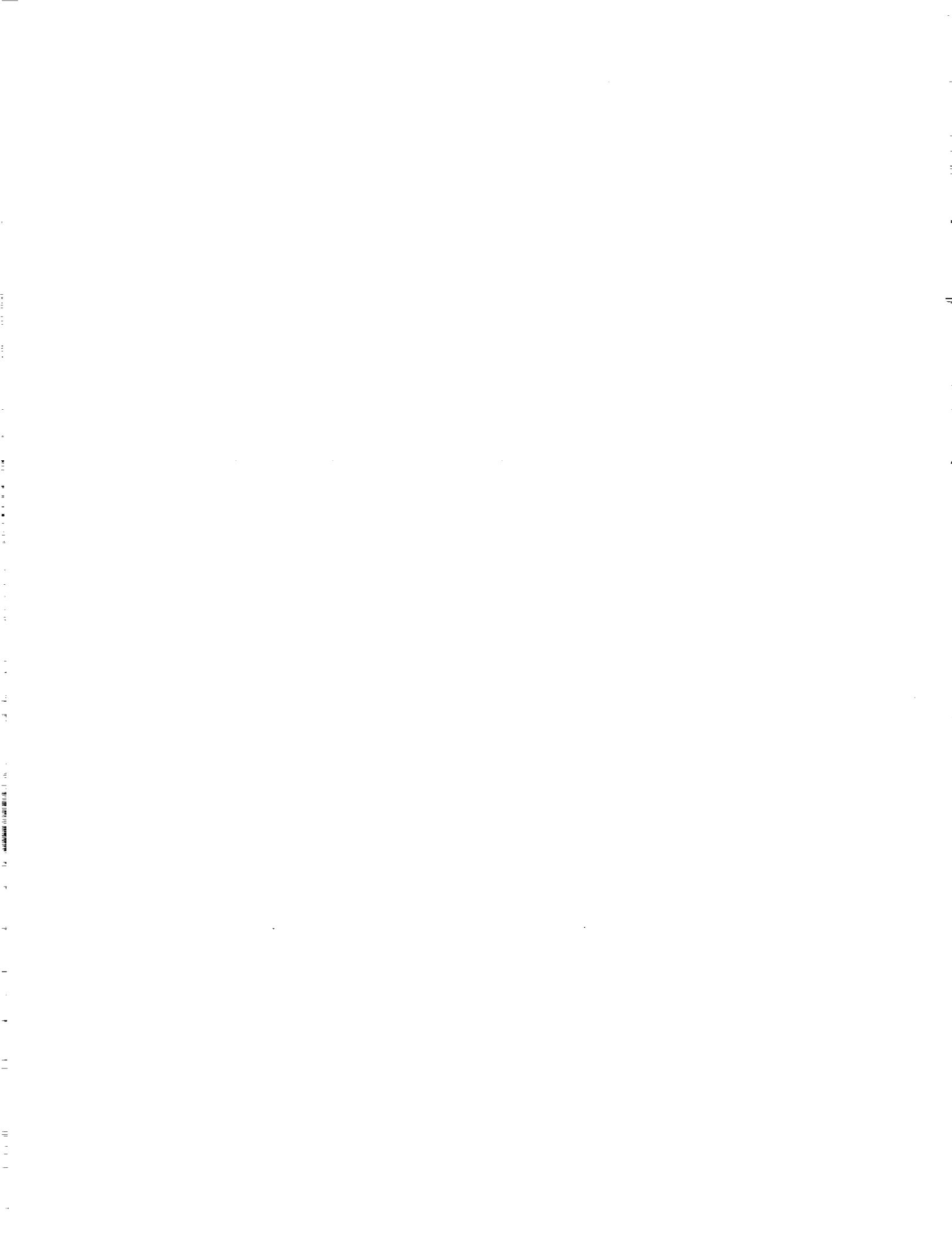
In the last section we have examined the property of goal-robustness as the last of the design specifications of an autonomous agent. This property has far-stretching consequences for spacecraft control. It means that ground control commands will not be unconditionally accepted, i.e. commands may be rejected when they don't meet the criteria specified above. The relation ground-control/spaceship becomes one of cooperation in which joint plan formation is possible by exchanging high-level goals and information. This is the way in which independent agents cooperate in a real-life situation (Werner, 1989, p. 7). This organizational structure is more appropriately classified as a MAS. Primarily this is the case because, due to communication limitation, the spacecraft can't 'think' on the ground and it has to take decisions directly in response to its environment (when the situation requires prompt action) or in absence of a consultant (when communication with ground control fails). In this light the expressed position "a fully autonomous space[craft] is neither achievable nor necessarily desirable." (Easter & Staehle, 1984, p. 5-2) has to be revised, while the question "... how long a space platform can perform a given function, even in the presence of new and existing faults, without intervention or direction from ground personnel or equipment" (Ibid.) can be answered by: indefinite, but more likely better if ground control recommendation is available. In that sense, as we have shown in this paper, we fully agree with the following two standpoints (Ibid. p. 3-4): "No autonomous system is actually free of human supervision; autonomous systems do not replace humans in this sense." because the high-level goals are always originating from humans and high-level goals can't be replaced

by the system itself; “[autonomous systems] provide much more flexibility for determining the optimal degree, nature and location of human participation in space activities”, indeed they do, because they determine the appropriateness of human advice and direction.

References

- Aarup, M., Munch, K. H., Fuchs, J., Hartmann, R., & Baud, T. (1994). Distributed intelligence for ground/space systems. In *Proceedings of the 3rd International Symposium on Artificial Intelligence Robotics, and Automation for Space* Pasadena, California, USA: JPL Publication 94-23.
- Bond, A. H. & Gasser, L., Eds. (1988). *Readings in Distributed Artificial Intelligence*, San Mateo, California. Morgan Kaufmann Publishers.
- Cariani, P. (1991). Emergence and artificial life. In C. G. Langton, C. Taylor, J. D. Farmer, & S. Rasmussen (Eds.), *Proceedings of the second workshop on artificial life*: Addison-Wesley.
- Castelfranchi, C. (1994). Guaranties for autonomy in cognitive agent architecture. In M. J. Wooldridge & N. R. Jennings (Eds.), *Proceedings of the 1994 Workshop on Agent Theories, Architectures and Languages* Chichester, West Sussex: John Wiley & Sons.
- Charniak, E. & McDermott, D. (1985). *Introduction to Artificial Intelligence*. Addison Wesley.
- Durfee, E. H., Lesser, V. R., & Corkill, D. D. (1987). Cooperation through communication in a distributed problem solving network. In M. N. Huhns (Ed.), *Distributed Artificial Intelligence* (pp. 29–58). San Mateo, California: Pitman Publishing\ Morgan Kaufmann.
- Durfee, E. H., Lesser, V. R., & Corkill, D. D. (1988). Coherent cooperation among communicating problem solvers. In (Bond & Gasser, 1988), (pp. 268–284).
- Durfee, E. H. & Rosenschein, J. S. (1994). Distributed problem solving and multi-agent systems: Comparisons and examples. In M. Klein & K. Sharma (Eds.), *Proceedings of the 13th International Distributed Artificial Intelligence Workshop* (pp. 94–104).
- Easter, R. W. & Staehle, R. L. (1984). *Space Platforms and Autonomy*. Technical Report JPL D-1973, Jet Propulsion Laboratory, Technology and Space Program Development, NASA/California Institute of Technology, Pasadena, California.
- Genesereth, M. R. & Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*, chapter 13, (pp. 307–327). Morgan Kaufmann Publishers: Los Altos, California 94022.
- Glisson, T. H. (1985). *Introduction to System Analysis*. New York: McGraw-Hill.
- Grant, T. J. (1994). Space/ground systems as cooperating agents. In J. Rash (Ed.), *Proceedings of the 9th Annual Goddard Conference on Space Applications of Artificial Intelligence* (pp. 357–368). Also to appear in: Telematics & Informatics.

- Koza, J. R. (1992). *Genetic Programming; On the Programming of Computers by Means of Natural Selection*. The MIT Press.
- Nilsson, N. J. (1994). Teleo-reactive programs for agent control. *Journal of Artificial Intelligence Research*, 1, 139–158.
- Owens, D. H. (1978). *Feedback and Multivariable Systems*. Southgate House, Stevenage, Herts, England: Peter Peregrinus ltd.
- Pidgeon, A. N. B., Seaton, G., Howard, G., & Peters, K.-U. (1992). *Spacecraft Autonomy Concept Validation by Simulation*. Phase 2 Final Report CR(P) 3604, Issue 1, European Space Agency.
- Rosen, R. (1986). On information and complexity. In *Complexity, Language and Life: Mathematical Approaches*. Springer Verlag.
- Steeb, R., Cammarata, S., Hayes-Roth, F. A., Thorndyke, P. W., & Wesson, R. B. (1988). Distributed intelligence for air fleet control. In (Bond & Gasser, 1988), (pp. 90–101).
- Werner, E. (1989). Cooperating agents: A unified theory of communication and social structure. In *Distributed Artificial Intelligence*, volume 2 (pp. 3–36). San Mateo, California: Morgan Kaufmann.



1995/20961

N95-27382

Integrated Planning and Scheduling for Earth Science Data Processing

Mark Boddy Jim White Robert Goldman
Honeywell Technology Center (HTC)
MN65-2200, 3660 Technology Drive
Minneapolis, MN 55418
{612-951-7403 | 7355 | 7336} Fax 612-951-7438
 {boddy | jwhite | goldman}@src.honeywell.com

57-82
47257
p. 11

Nick Short, Jr.
ISTB, Code 935
NASA Goddard Space Flight Center
Greenbelt, MD 20771
{short@unloggin.gsfc.nasa.gov}

Abstract

Several current NASA programs such as the EOSDIS Core System (ECS) have data processing and data management requirements that call for an integrated planning and scheduling capability. As we have shown in previous work, the scale and complexity of data ingest and product generation for ECS will overwhelm the capabilities of manual planning and scheduling procedures. Meeting this challenge requires the innovative application of advanced technology. Some of our work on developing this technology was described in a paper presented at the 1994 Goddard AI Conference, in which we talked about advanced planning and scheduling capabilities for product generation. We are now moving to deploy some of the technology we have developed for operational use.

We have implemented a constraint-based task and resource scheduler for the GSFC Version 0 (V0) Distributed Active Archive Center (DAAC) requirements. This scheduler, developed by Honeywell Technology Center in cooperation with the Information Science and Technology Branch and with the V0 DAAC, makes efficient use of limited resources, prevents backlog of data, and provides information about resource bottlenecks and performance characteristics. It handles resource contention, prevents deadlocks, and makes decisions based on a set of defined policies. The scheduler efficiently supports schedule updates, insertions, and retrieval of task information. It has a graphical interface that is updated dynamically as new tasks

arrive or existing tasks are completed. The kernel scheduling engine, called Kronos, has been successfully applied to several other domains such as space shuttle mission scheduling, demand flow manufacturing, and avionics communications scheduling. Kronos has been successfully applied to scheduling problems involving 20,000 tasks and 140,000 constraints, with interactive response times for schedule modification on the order of a few seconds on a SPARC10.

In this paper, we describe the experience of applying advanced scheduling technology operationally, in terms of what was accomplished, lessons learned, and what remains to be done in order to achieve similar successes in ECS and other programs. We discuss the importance and benefits of advanced scheduling tools, and our progress toward realizing them, through examples and illustrations based on ECS requirements. The first part of the paper focuses on the Data Archive and Distribution (DADS) V0 Scheduler described above. We then discuss system integration issues ranging from communication with the scheduler to the monitoring of system events and re-scheduling in response to them. The challenge of adapting the scheduler to domain-specific features and scheduling policies is also considered. Extrapolation to the ECS domain raises issues of integrating scheduling with a product-generation planner (such as PlaSTiC), and implementing conditional planning in an operational system. We conclude by briefly noting ongoing technology development and deployment projects being undertaken by HTC and the ISTB.

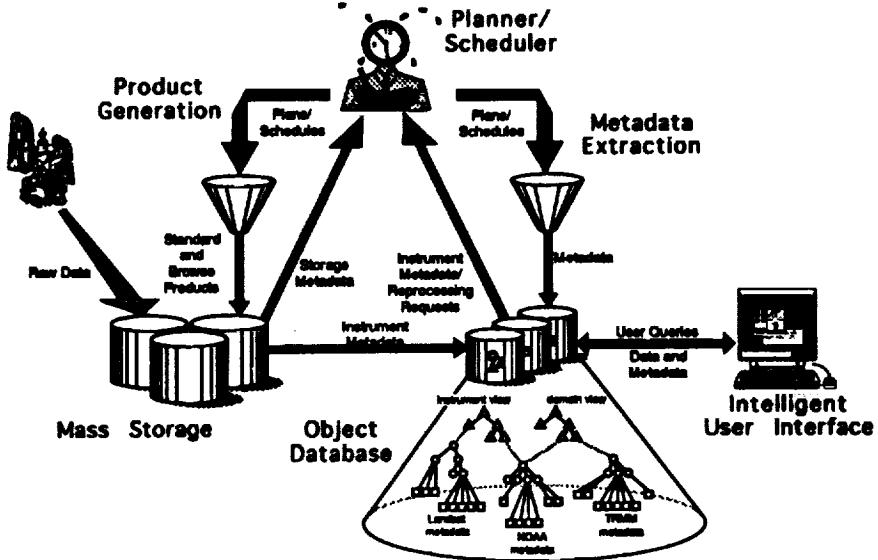


Figure 1: The IIFS

1 Introduction

In both joint and separate work at NASA's Goddard Space Flight Center and the Honeywell Technology Center, we have been working on automating the acquisition, initial processing, indexing, archiving, analysis, and retrieval of satellite earth science data, with particular attention to the processing taking place at the DAACs.

After describing our motivation in section 2 and related work in section 2.1 and section 3 we focus on the DADS V0 Scheduler. In section 4 we present general scheduling requirements initially derived from the DADS application, but extended to encompass similar NASA instrument processing such as the Clouds and Earth's Radiant Energy System (CERES) and the Moderate Resolution Imaging Spectroradiometer (MODIS). Then, we describe the implementation and operation of the prototype DADS V0 Scheduler with particular attention to lessons learned that have enhanced its generality and reusability for other applications. We conclude with a short summary of our conclusions and plans for future work.

2 Motivation

Management of complex systems requires skill in a variety of disciplines. Two critical management disciplines involve deciding what activities to perform, which we call *planning*, and deciding when those activities should be performed, which we call *scheduling*. In large systems such as ECS these functions must be automated, since the sheer volume of data will over-

whelm human managers.

It is not sufficient to simply plan and schedule the required activities. These decisions inherently model the target system. Even when this model is made highly detailed, it can never capture all of the details and possible future behaviors of an actual system. Scheduled activities will require more or less time than scheduled. Requests will arrive unexpectedly. Resources will be unavailable or will fail during use. Efficient operation and resource utilization requires that execution must be monitored and future activities rescheduled in response to real world events.

Hence, the overall advantages for using scheduling include:

- automation of routine operations,
- timely delivery of data products,
- efficient use of computational resources.

Satisfaction of these requirements will lead to reduction in staff, use of cheaper hardware, and user satisfaction. These principles are being applied to both the Intelligent Information Fusion System (IIFS) and the DADS in the next sections.

2.1 Intelligent Information Fusion

Since 1989, the IIFS is a prototype system for testing advanced technologies for processing, archiving, and retrieval of remote sensing imagery. The IIFS is currently being applied to the next generation direct-readout domain, whereby data are received from the

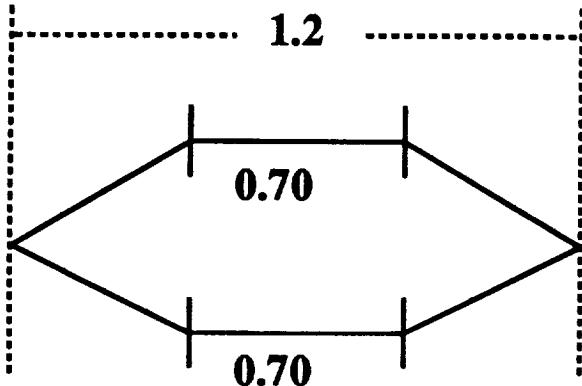


Figure 2: A Simple Problem with Duration and Partial Orders

direct broadcast from orbiting platforms for the region encompassing the acquisition and loss of the spacecraft's signal. These inexpensive ground systems are used for weather forecasting in remote areas of the world, collection of *in situ* data, and calibration/validation of sensor data to name a few.

The planning/scheduling portion of the IIFS is used to manage the production pipeline. Essentially, the planner/scheduler sacrifices accuracy for time in generating the data product. That is, if enough time and resources exist, then the planner/scheduler generates the normal data product, called a standard product in the EOS nomenclature. If not, then the planner substitutes computationally cheaper algorithms until resource constraints can be met. The result is called a browse product and is used solely by the scientists during the data selection phase. Should the scientist decide that greater accuracy is required, then the plan can be regenerated under less computationally constrained circumstances. This may, for example, involve issuing a request to EOSDIS' DADS if the direct-readout center is incapable of handling the request.

During the past few years, several planner/schedulers have been tested in this domain. The next section briefly discusses only one of these planners, called PlaSTiC.

3 PlaSTiC

PlaSTiC (Planning and Scheduling Tool implemented using Constraints) is an automated planning tool designed to automatically generate complex plans. PlaSTiC was developed as a prototype for the generation image analysis plans (browse products and scientific data) in the EOSDIS domain.

A typical plan might detail the processing steps to be

taken to clean up, register, classify, and extract features from a given image. Plan steps will be executed in a resource-limited environment, competing for such resources as processing time, disk space, and the use of archive servers to retrieve data from long-term mass storage. Choices of these algorithms depends on the type of satellite, region of the country, computation characteristics (e.g., deadline, resource requirement, etc.).

PlaSTiC is an integration of hierarchical planning and constraint-based scheduling. TMM provides the basis for temporal reasoning and constraints. The planning component is based on an implementation of NON-LIN developed at the University of Maryland. PlaSTiC extends NONLIN-style planning to include reasoning about durations and deadlines.

The schemas used by PlaSTiC, which are based on NONLIN's Task Formalism (TF), have been extended to record information about the estimated and worst-case duration of a given task, and about the task's resource usage. This information is used during plan construction, for example in the rejection of an otherwise promising expansion for a given sub-task because it requires more time than is available. It is also used in the construction of detailed schedules for image processing tasks.

The fact that actions take time was abstracted out in the earliest domain models. Planners using these models will be of limited use in domains where synchronization with other events or processes is important. This may include such domains as manufacturing planning and scheduling, spacecraft operations, robot planning in any but the most simplified domains, and scheduling distributed problem-solving or other processing. It certainly includes analysis and retrieval planning within ECS.

Several planners include representations for metric time and action durations. This kind of reasoning tends to be computationally expensive. Forbin, Deviser, and Sipe all suffer from performance problems limiting the size of the problems to which they can be applied. Oplan-2 appears to be able to handle somewhat larger problems than the other planners mentioned here.

Implementing an efficient temporal reasoning system is not the sole hurdle, however. Adding duration to nonlinear plans increases the difficulty of determining whether or not the current partial plan can be refined into a plan that will have the desired effects. In fact, it becomes difficult to determine simply whether the actions described in the current partial plan can even be executed.

Consider the simple plan fragment in Figure 2. There are two unordered tasks, each annotated with an estimated duration. If actions can only be taken in sequence, the two tasks depicted must eventually be ordered. When the planner tries to order them, it will discover that neither ordering will work, because there simply isn't room for them to be performed in sequence. In general, determining whether there is an ordering for a set of actions constrained in this way is a hard problem.

To date, two methods have been used to address this problem. The first is *simulation*: the planner maintains a partial order, and after every modification expends some effort exploring the corresponding set of total orders to ensure that there is some feasible total order [Miller, 1985, Muscettola, 1990]. As generally employed, this is a heuristic method: the planner gives up before exploring the complete set of consistent total orders. Another approach, described in [Williamson and Hanks, 1988], involves organizing a partially-ordered plan into a tree of abstract operator types, known as *Hierarchical Interval Constraints* (HIC). Each HIC type has a function defined for calculating bounds on its duration. For the example in Figure 2, the two activities would be contained in an HIC whose duration was calculated by summing the duration of the included operators. The problem with this approach is the required tree structure. If actions must be ordered for reasons that are not locally determinable (e.g. because of resource conflicts, not because they are sequential steps in some task reduction), this representation will break down. It may be possible to augment Williamson and Hanks' representation to cope with a limited number of special structures representing such nonlocal information.

In PlaSTiC, we have started with the assumptions that resource conflicts are significant, that activity durations are nontrivial, and that deadlines will be a factor. For these reasons, the temporal reasoning underlying PlaSTiC is implemented in a full-fledged scheduling engine, so that resource conflicts can be noted and resolved as part of the planning process. Similarly, deadline checks are performed automatically as task reduction and order proceeds, triggering backtracking as necessary. The task hierarchy employed by PlaSTiC maintains at all levels a set of duration estimates, so that deadline and resource conflicts may be noticed before a task is expanded all the way to primitive actions. This approach is consistent with the simulation-based technique described above, but so far we have had considerable success in simply resolving possible problems (e.g., potential resource conflicts) as they arise.

The scheduling component of PlaSTiC is built on the Kronos scheduling engine. The DADS V0 Scheduler

described below employs this same technology, but with significant extensions to address domain specific scheduling and system integration issues.

4 DADS V0 Scheduler

Unlike direct-readout centers which will dynamically create data flow sequences, the DADS of EOSDIS maintains a database of fixed data flow diagrams. These are retrieved upon request from a database to accomplish various DADS functions. Hence, the DADS required only scheduling and dispatch technology for nominal operations.

In particular, the DADS V0 Scheduler is responsible for scheduling actions and resources to ingest data from a network to buffer disks, transfer buffered data to a mass storage archive, and to retrieve archived data upon request. The scheduler was developed concurrently with the design and implementation of the GSFC V0 DADS. Consequently it was essential that the architecture and interfaces be able to tolerate changes as the system design evolved. The baseline architectural environment of the scheduler is depicted in Figure 3. This environment continues to evolve, but its conceptual and functional characteristics remain stable, so many system changes can be accommodated in the Application Program Interface (API).

The DADS Manager submits scheduling requests, handles errors, and retrieves schedule information. The Task Dispatcher periodically queries the scheduler for a list of upcoming scheduled activities to be executed. The execution monitor notifies the scheduler of events that affect the schedule.

4.1 Approach

The scheduling tool described in this paper was designed to meet the scheduling and resource allocation needs of the GSFC V0 DAAC while simultaneously using the IIFS as a testbed.

Constraint envelope scheduling technology offers an attractive, proven method of meeting the scheduling needs of data archiving and distribution. This technology, embodied in Honeywell's enhanced implementation of the Time Map Manager (TMM), supports the concept of a Temporal Constraint Graph (TCG) which can be used to represent multiple projections of future system behavior, thereby providing rapid rescheduling with minimal disruption in the presence of schedule uncertainty.

The DADS V0 Scheduler is an application of the Kronos scheduling engine, built on top of TMM. Kronos has been successfully applied to domains such as

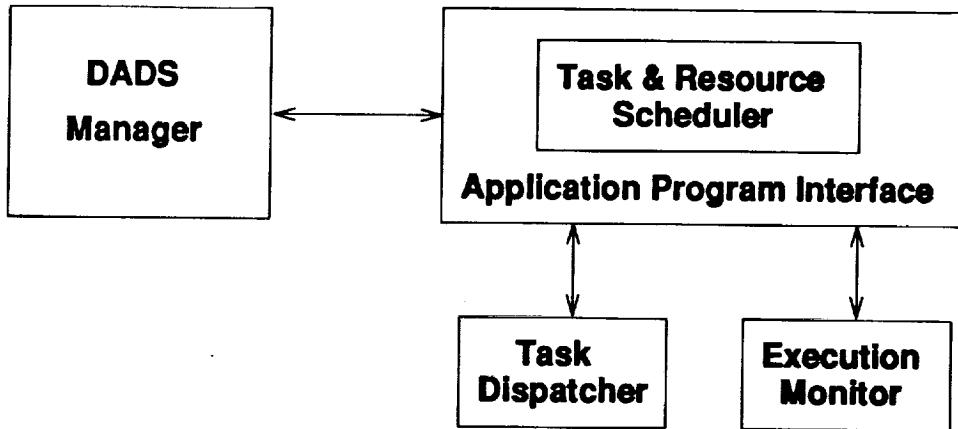


Figure 3: The DADS VO Scheduler's Architectural Environment

space shuttle mission scheduling, demand flow manufacturing, and avionics communications scheduling. It has handled scheduling problems involving 20,000 tasks and 140,000 constraints, with interactive response times for schedule modification on the order of a few seconds on a SPARC10.

4.2 Scheduler Requirements

Detailed scheduler requirements were initially established for the DADS application, then extended and adapted to encompass the scheduling needs of other NASA programs. The following paragraphs summarize requirements at a high level. They confirm the need to be appropriate to the application domain, to be compatible with the target system, and to provide responsive performance reliably.

Domain Appropriate - Commercial scheduling tools sacrifice domain relevance to extend their range of applicability, and hence their marketability. They often lack the capacity to efficiently handle the precise scheduling needs of large, complex applications. In order to select or define a scheduling tool that is domain appropriate, application driven requirements must be established. Whenever possible, these requirements should be based on multiple examples of domain operations and scheduling functions using realistic data sets. They must include quantitative demonstration that capacity and performance goals can be met simultaneously.

Since the GSFC V0 DADS is being developed concurrently with the prototype scheduler, we were careful to maintain a high degree of generality in the scheduler implementation. By first building a core scheduling capability derived from our Kronos scheduling engine, and then extending that capability through specialization, we were able to meet the specific needs of DADS

while providing a scheduling tool that can easily be applied to similar problem domains.

Stated as a system requirement, the scheduling core domain model must be compatible with objects and functions required by the target application. Further, its customization capabilities must support accurate modelling of every schedule relevant aspect of the domain. Care should be taken to ensure that this model reflects the intended scheduling policies and procedures of the application, and not the characteristics of analytical models used to project system performance.

Details of the scheduling core domain model are described in section 4.4.1. For the prototype scheduler, subclasses were created to capture application specific attributes and relationships. These attributes may be used to carry system data through the schedule or to support performance monitoring and analysis.

In one instance this derivation was particularly enlightening. The Kronos scheduling engine associated resource utilization with the duration of the activities to which a resource was assigned. If a common resource was required by multiple disjoint activities, it was expected that an encompassing parent activity would specify the requirement and would be assigned the shared resource. In the GSFC V0 DADS, there is no encompassing parent activity. Resource utilization can be initiated by one activity (e.g., through transfer of network data to a space on buffer disk) and must persist indefinitely into the future (e.g., until a future activity transfers it to the archive).

By creating persistent requirement and persistent resource profile classes as subclasses of the requirement class and resource profile class, respectively, we were able to provide the necessary scheduler functionality with a minimum of disruption. Persistent requirements have the option of specifying that they begin,

use, or ending with their associated activity. This allows the resource allocation to be open ended if desired.

To be effective, any tool must be functionally complete. That is, it must be able to solve the problems to which it is applied. A scheduler must enforce structural constraints (i.e., predecessor-successor and parent-child relationships), temporal constraints (e.g., earliest start or deadline), and resource availability constraints while carrying out the desired scheduling and resource allocation policies in an automated fashion. In the prototype scheduler, policies are currently encoded as functions and a domain specific algorithm (as described in section 4.4.3). We plan to eventually excise policy details from the scheduler by defining syntax for policy specification. This specification will then be input to the scheduler and used to control scheduling and resource allocation decisions.

Compatible - The scheduling tool described here is designed be integrated as a functional component into the target application system. It cannot dictate requirements to that system, rather, it must adapt to the physical and logical demands of the encompassing system. The scheduler must execute on available hardware running the specified operating system. It must be able to communicate with asynchronous functional modules of application system via standard interprocess communication system facilities.

The scheduler must also be linguistically compatible with the surrounding system. It must be able to interpret and respond appropriately to requests for service and information. The prototype scheduler meets this requirement in several ways. The scheduler includes an API customized to the syntactic and semantic needs of the DADS modules with which it interacts. An underlying set of basic API functions facilitates this customization.

The scheduler supports the notion of activity state. The exact states and legal state transitions are defined for the application. In DADS, activities can be scheduled, committed, dispatched, executing, complete, or failed. Additional states and even additional state dimensions can be added as the need arises.

Responsive - Performance is often a critical requirement, but it is frequently overlooked in scheduling. It is assumed that scheduling will be performed once in an initial scheduling effort and that the resulting schedule will satisfactorily describe the actual execution of activities. This view is seldom correct.

We have segregated the total problem into two phases, planning (what to do) and scheduling (when to do it).

By making this distinction, we have not only, made each aspect more manageable, but we can tailor the functionality and performance of each component's implementation to the needs of the application. Planning typically occurs before scheduling, though replanning may become necessary. In the GSFC V0 DADS application, there is a small set of functions to be performed (e.g., ingestion, distribution). These can be pre-planned in advance and described to the scheduler as tasks (with subtasks).

The scheduler must, on demand and in near real time, fit each new instance of a task into the current schedule in accordance with task priorities and deadlines while ensuring that necessary resources will be available. As actual events occur in the execution of the scheduler, it must rapidly reschedule to reflect the impact of the event. It must provide data to support graphic presentation of the current schedule, and even allow operator manipulation of tasks.

Reliable - The fault tolerance approach employed by the target application must be supported by the scheduler. In the GFSC V0 DADS this translates to requirements for redundant archiving of schedule information and rapid recovery of the schedule after a failure. The prototype scheduler does not fully include these features at present. However, basic mechanisms needed for reload are present in the script processor described in section 4.3. Also, previous schedulers based on the Kronos engine have included schedule storage and reload capabilities.

4.3 Prototype Environment

The DADS V0 Scheduler is being developed concurrently with the GSFC V0 DADS. Consequently it was necessary to provide a stand-alone environment in which to test and demonstrate scheduler functionality. The operation of components external to the scheduler was simulated via a script processor as shown in Figure 4. The script processor is controlled from a demonstration Graphical User Interface (GUI) that displays schedule activities and resource utilization profiles. Snapshots of the demonstration GUI screen may be seen in Figures 7 and 8. The GUI supports selection and execution of an event script which the script processor translates into API commands that it sends to the scheduler.

A typical script initializes the scheduler by describing the resources available for scheduling, commands the creation of activities to be scheduled, and simulates execution events such as completion of execution. The script also notifies the GUI as objects to be displayed are created.

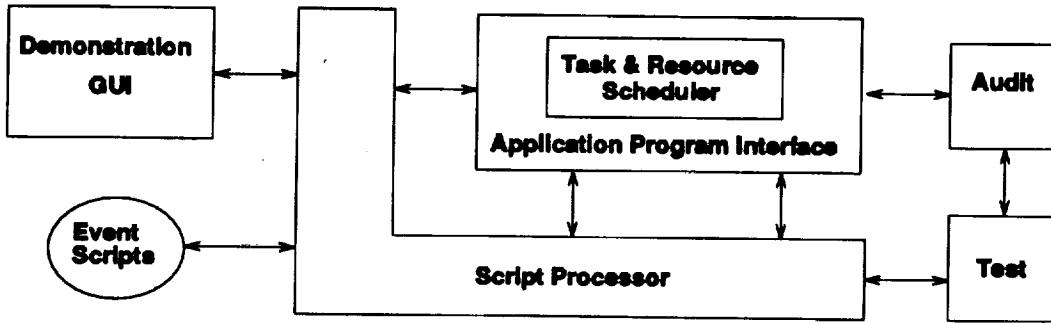


Figure 4: The Prototype System Architecture

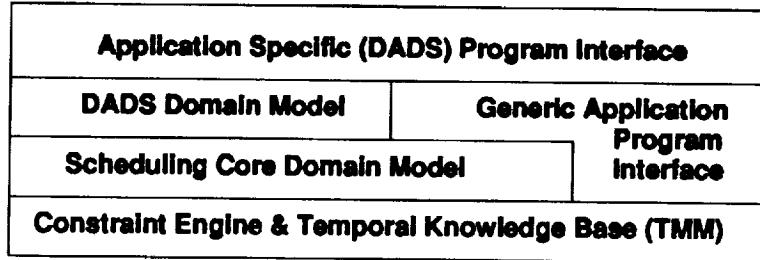


Figure 5: The Architecture of the Scheduler

Graphical presentation of scheduler operation is visually convincing, but it is inconvenient for testing and benchmarking purposes. Recently, auditing and test functions were added to facilitate execution and validation of complex event scripts. The test function automates the execution of scripts and the invocation of the audit function, which checks the schedule for consistency and correctness.

4.4 Architecture of the Scheduler

The internal architecture of the scheduler is depicted in Figure 5. The base layer supplies basic temporal reasoning capability. This includes objects such as uncertain time-points and constraints, and functions for updating and querying the temporal knowledge base.

The Scheduling Core Domain Model supplies the basic objects and functions needed for scheduling and resource management. Combined with the Generic API, these layers form a core scheduling capability that can be applied to various scheduling domains. In the DADS V0 Scheduler implementation, the base domain model was extended through specialization and extension to provide appropriate domain-specific capabilities, shown in the figure as the DADS Domain Model and the DADS API.

4.4.1 Domain Model

Key object classes of the scheduling core domain model include resources, requirements, activities and hierarchical activities. These are shown in Figure 6 along with related objects classes of the DADS scheduling domain model.

An activity represents an action to be scheduled. Each activity has an associated main-token which defines its end points in time and its possible duration range. An activity may be linked to multiple resource requirements. These abstractly define attributes that must be satisfied by the resources allocated to the activity. A subclass of the activity allows hierarchical activity structures to be defined. These were used in the DADS scheduler to implement tasks with component subtasks.

As an example, in the DADS application, a data ingestion task will have several subtasks. The data buffering subtask requires access to the FDDI network and a specific amount of space on one of the data ingestion magnetic disks. A subsequent archiving subtask requires access to the data on buffer disk and space on the UNITREE archive magnetic disk.

The core resource classes allow resources to be conceptually organized into pools using a hierarchical name structure (which permits wildcards) and using a list of resource attributes. Each resource has an associated availability that defines the maximum quantity of that

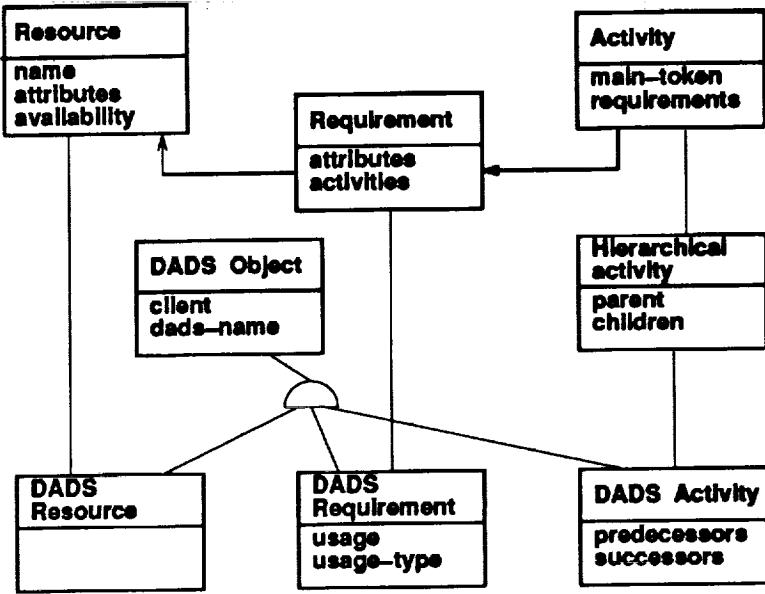


Figure 6: Key DADS Scheduling Object Classes

resource and its temporal range.

Specializations of the core object classes extend the hierarchy to include characteristics of the target domain. In the DADS scheduler these specializations share a common parent class, the DADS object, which defines attributes every DADS activity, resource requirement, or resource must have. Only the client and dads-name attributes are shown in the figure.

4.4.2 Application Program Interface (API)

The Application Program Interface was specified formally by documenting data content (i.e. fields and forms) of the primary information components (i.e. tasks, subtasks, resources, etc.) exchanged between the scheduler and DADS subsystems. For each command, the documentation details the participants in the exchange utilizing the command, the conditions under which the command occurs, the intent (semantics) of the command, and the scheduler's response to the command under both normal and error conditions.

The following command categories describe the functions of the scheduler visible via the API. The categories have been intentionally kept rather abstract and high level here. Not all command categories have been fully implemented in the prototype scheduler.

Definition/Instantiation - Inform the scheduler of the existence of scheduling entities such as activities (i.e. tasks and subtasks), resources, and abstract resource utilization requirements. These commands do not cause scheduling to occur.

Modification - Change the specifics of information known to the scheduler. This category encompasses only changes to the scheduling problem (e.g. relaxation of a deadline). It does not include notification of real-world execution events.

Interrogation/Retrieval - Retrieve schedule and resource allocation information from the scheduler. This information is based on the scheduler's model of the problem space, its record of past events, and its projection of future events including resource utilization.

Scheduling/Rescheduling - Compute a new schedule with resource allocations. Commands in this category may be invoked indirectly by commands in the Update/Synchronization category.

Update/Synchronization - Inform the scheduler of the occurrence of real-world events (e.g. activity execution completion) which may affect the schedule. This category also includes commands for the transfer of responsibility for an activity from the scheduler to another subsystem (e.g., an execution monitor or dispatcher).

Notification - Inform another subsystem that a problem (or potential problem) has been detected by the scheduler.

Communication Handshaking - Provide positive acknowledgement of information transfer.

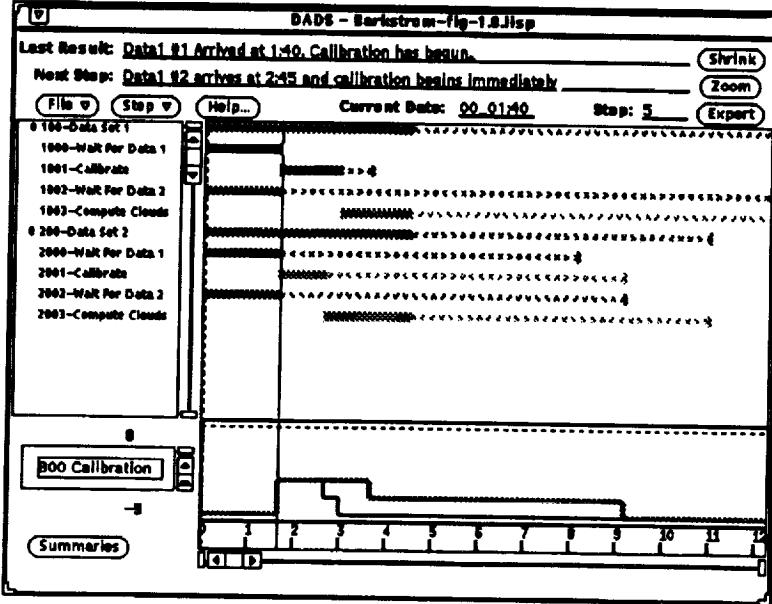


Figure 7: Schedule after Data 1 #1 Arrives

Fault-Tolerance/Recovery - Support for information backup and recovery from failures.

4.4.3 Scheduling Policy

The operation of the scheduler is controlled by scheduling policies. These are currently captured in domain specific algorithms for resource assignment and activity scheduling.

The baseline resource assignment and scheduling algorithm is:

For each activity to be scheduled:

- If the activity has component activities, Schedule each of its component activities (i.e., apply this algorithm recursively).
- If the activity is scheduleable,
 - For each resource requirement of this activity:
 - If a satisfactory resource is available for use without causing it to be oversubscribed, assign that resource to meet the requirement. Availability implies that the resource is part of the resource pool specified in the resource requirement and has the attributes specified in the resource requirement.
 - If no satisfactory resource is available, apply the following stratagems in sequential order, using the possible resources until one of them successfully eliminates the

oversubscription:

- * Constrain the order of activities involved in the oversubscription:
 - Individually *before* the activity, or
 - Individually *after* the activity, or
 - Collectively *before* the activity, or
 - Collectively *after* the activity.
- * Relax the deadline of activities involved in the oversubscription and constrain the order of activities (as above)
- * Constrain the order of parent activities of the activities involved in the oversubscription (as above)
- * Report failure [and Exit]
- If the activity is still scheduleable and all component activities of this activity have been scheduled, Mark the activity scheduled.

Then update:

- The schedule's temporal knowledge base,
- The time bounds of all changed resource utilization profiles.

4.5 Scheduling Example

The operation of the prototype scheduler is revealed in Figures 7 and 8. In this simple example, taken from the Clouds and the Earth's Radiant Energy System (CERES) domain, two instances of a single task type

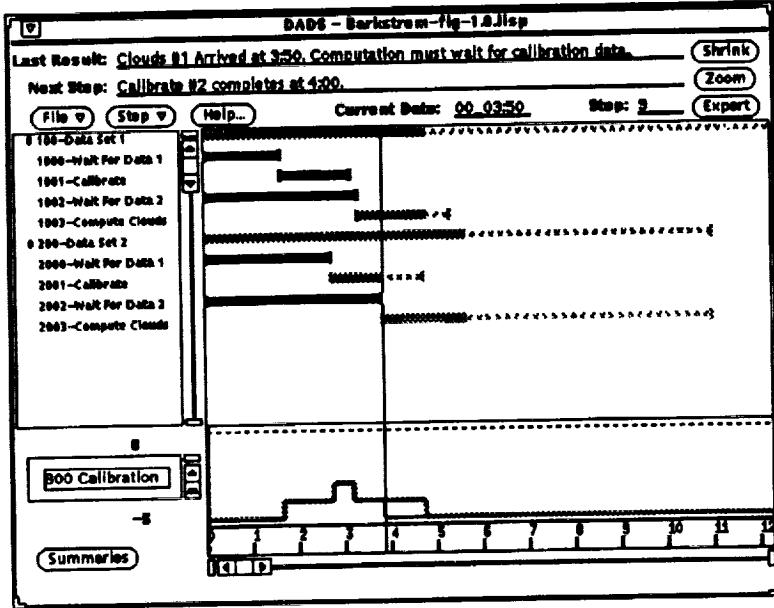


Figure 8: Schedule after Data 2 #2 Completes Execution

have been scheduled. Each task consists of four related subtasks with interdependencies. The first subtask is to wait until a particular radiation budget data set arrives. The second subtask is to calibrate and Earth-locate that data set. A calibration resource is required by this subtask. The third subtask is to wait for a corresponding cloud identification data set. The final subtask is to compute cloud data by combining the calibrated radiation budget data with the cloud data to produce a combined data product. The Calibrate subtask cannot occur until its Data 1 is available. The Compute Clouds subtask cannot occur until the Calibrate subtask is complete and its Data 2 is available. For illustrative purposes, the second task has been given a deadline of 11:00 while the first task has no deadline.

Figure 7 shows the situation after the first dataset arrives. The earliest scheduled time for each activity is shown to the right of its name as a solid horizontal bar. Dashed lines indicate the range of possible occurrences of the activity. The current time is represented as a vertical line.

Subtask 1001 has now started because subtask 1000 has finished. Subtask 1003 cannot start until subtask 1001 completes. Subtask 2001 could start immediately, but since its predecessor subtask, 2000, is still executing, it will slip as time passes. Because of a similar predecessor dependency on subtask 2001, subtask 2003 will also slip. The scheduler automatically reschedules the earliest start and earliest end times of these activities as time passes.

The resource utilization profile of one of the resources used by the example activities is shown at the bottom of Figure 7. The profile indicates both the scheduled (black) and potential (gray) utilization of the resource. The API of the DADS V0 Scheduler provides query commands for determining the relationships between resource utilization and scheduled activities, but in this example careful examination of the shape of the profile reveal that increments of the Calibration tool resource have been allocated to satisfy the requirements of subtasks 1001 and 2001.

At a later time, after more of the subtasks have completed execution, the situation is noticeably different. This is shown in Figure 8. Subtask 1003 did not start immediately after Subtask 1001 (Calibrate) because of its additional dependency on the completion of subtask 1002 (Data 2). Notice that although task 100 has no deadline, a maximum end time for subtask 1003 has been scheduled because that subtask has an associated maximum duration.

The resource utilization profile for the Calibration tool resource has changed significantly from that projected in Figure 7. This is because the start of subtask 2001 could not be predicted reliably because of its dependency on the completion of subtask 2000. The execution of subtask 2001, and the utilization of the Calibration resource was rescheduled until its Data 1 arrived.

Even this simple example shows that accurate scheduling and optimization of resource usage requires a scheduling tool that can rapidly reschedule future activities in response to real-world events.

5 Summary, Conclusions and Future Work

In this paper, we have presented results of the application of Honeywell's scheduling technology to an application of data archiving and distribution. We have described our progress to date and some insights regarding further application of this technology to other domains. Moving to broader operational use will require further refinement and development.

We plan to continue development and refinement of the planning and scheduling capabilities described in this paper. Our efforts will be focused on increasing their applicability and achieving the goal of realization of the Intelligent Information Fusion System. In the near term we will provide documentation, training, and support materials in order to obtain design feedback through use of these tools. We will simultaneously continue to extend their functionality in support of additional application domains.

References

- [Miller, 1985] David P. Miller. Planning by search through simulations. Technical Report 423, Yale University Computer Science Department, 1985.
- [Muscettola, 1990] Nicola Muscettola. Integrating planning and scheduling to solve space mission scheduling problems. In *Proceedings DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*, pages 220–230. Morgan Kaufmann, November 1990.
- [Williamson and Hanks, 1988] Mike Williamson and Steve Hanks. Efficient temporal reasoning for plan projection. In James Hendler, editor, *Proceedings of the First International Conference on Artificial Intelligence Planning Systems*, pages 313–314, 1988.



A Sustainable Genetic Algorithm for Satellite Resource Allocation

47258
P. 15

R J Abbott

*The Aerospace Corporation
and
California State University,
Los Angeles*

M L Campbell

The Aerospace Corporation

W C Krenz

The Aerospace Corporation

A hybrid genetic algorithm is used to schedule tasks for a satellite, which can be modelled as a robot whose task is to retrieve objects from a two dimensional field. The objective is to find a schedule that maximizes the value of objects retrieved. Typical of the real-world tasks to which this corresponds is the scheduling of ground contacts for a communications satellite.

An important feature of our application is that the amount of time available for running the scheduler is not necessarily known in advance. This requires that the scheduler produce reasonably good results after a short period but that it also continue to improve its results if allowed to run for a longer period. We satisfy this requirement by developing what we call a sustainable genetic algorithm.

1.0 Introduction

Planning, i.e., deciding in advance on a course of action, is a long-standing and difficult problem.[Al90] Planning for a mobile robot is yet more complex as a result of the interactions among robot dynamics, horizon planning, and task valuation. When expressed numerically, planning problems are frequently ill-conditioned due to the combination of continuous

and discrete variables over which decisions must be made. Consequently, not only are standard optimization techniques (such as non-linear programming) time consuming, they generally fail to provide satisfactory results. A good robot planning algorithm should:

- a) generate efficient schedules of actions;
- b) abide by system-imposed constraints;
- c) be flexible enough that different tasking goals can be realized without major implementation changes;
- d) be implemented in an efficient computational framework either through an extremely fast serial algorithm or through parallelization;
- e) provide reasonable schedules on a short timeline and better schedules when more time is available.

This paper discusses a genetic algorithm approach to robot scheduling. The remainder of the paper is organized as follows. *Problem Statement* describes the robot's problem. *Analysis for a Numerical Approach* describes the equations that would have to be maximized were the problem to be solved numerically. *A Brief Introduction to Genetic Algorithms* provides an introduction to genetic algorithms. *The Application of Genetic Algorithms to Object Retrieval* describes our use of genetic algorithms for this problem. *Diversity Management and Sustainable Genetic Algorithms*

describes our techniques for allowing our genetic algorithm to find good results without population convergence. *Results* describes our results. *Future Work* describes future directions. *Conclusions* discusses conclusions and further work.

2.0 Problem Statement

The problem under consideration can be modelled as the scheduling of a robot whose job it is to retrieve objects while traversing a field. The robot moves at a constant rate along a set of pre-assigned horizontal passes. The robot has an arm and a hand. The arm has a limited length (both minimum and maximum) and a limited angle at which it can operate. (Details of robot kinematics are not addressed in this paper.)

Each object has a value and a location in the field, both known in advance. To retrieve an object, the robot hand must stay at the object's location for a given time. The objective is *to retrieve the highest total value of objects*.

Additional constraints may be imposed which make the problem even more complex.

- The value of an object may depend on which other objects are retrieved.
- The robot may have a net (instead of a hand) on the end of its arm with which it may retrieve many objects at once.
- The time required to retrieve an object may be expressed as a function of the robot's arm geometry. For instance, it may take more time to retrieve an object if the arm has to reach further or if it needs to deviate from its ideal position.
- The field may contain widely-spaced objects that need to be retrieved within some specified planning horizon.
- Some of the objects may have very precise angles at which they must be accessed.

- There may be a limit to the length of time the robot may operate during each pass. The limit may derive from physical constraints on the robot's operation, such as power or thermal limitations.

3.0 Analysis for a Numerical Approach

Initial attempts to produce efficient schedules were developed in a traditional optimization framework. Figure 1 shows a test suite of data that was constructed to illustrate the problem. A random field of 100 objects was created, and the robot is given 3 parallel passes through the field. In this example, three different kinds of objects are present. They are represented by different symbols for the different values.

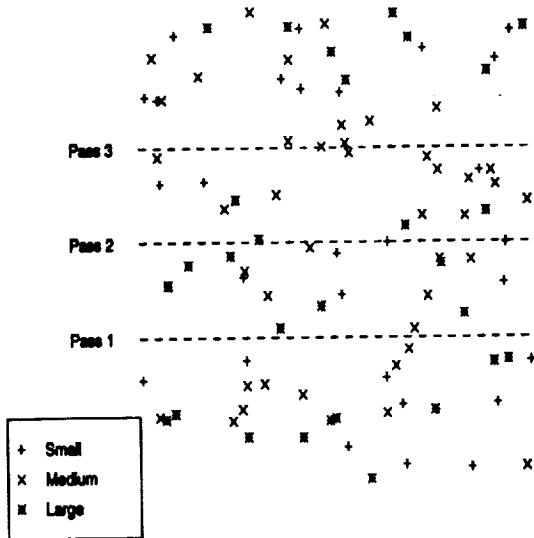


FIGURE 1. A testbed

The time between the completion of retrieving one object and the start of retrieving the next object must be greater than the time required to move the arm from one object to the next. The arm dynamics here are modeled simply as a fixed maneuver rate times the distance between subsequent objects:

$$T_{Man}(x_i, x_{i-1}) = \text{ManRate} \|x_i - x_{i-1}\| \quad (1)$$

where the variables are defined as:

T_i	Time required to retrieve an object and move to the next one
$T(x_i)$	Time at which object x_i is retrieved
$T_{Pickup}(x_i)$	Time required to retrieve object x_i
$T_{Man}(x_i, x_{i-1})$	Time required to move from one object to the next
ManRate	Maneuver rate of robot arm (distance/time)
\dot{x}_i	Vector position of point x

More complicated dynamics could be modeled without loss of applicability of the fundamental algorithms.

An added complexity is that the robot arm length and pass coordinates are such that some of the objects may be retrieved from any of a number of passes through the field. Objects in the center of the field, for example, are accessible from each of the passes; objects at the edge are accessible from only one pass.

The problem can be cast as an optimization problem (maximize the value of the objects retrieved) subject to the above constraints on robot reach and dynamics. Mathematically, this is cast in standard optimization form as:

$$\max_{t_x} f(x) : g(x) \leq 0 \quad (3)$$

where the parameter t_x over which the set will be optimized is the set of scheduled times for each point 'x.' The attributes of each point include:

$$x = \begin{bmatrix} \text{Value} \\ \text{Position} \\ \text{Accesses} \\ \dots \end{bmatrix} \quad (4)$$

where the value determines which objects are more important, the position drives dynamics constraints, and the number of accesses to an object may be used to decide which objects may be more easily postponed to a later pass.

Other attributes are also possible (for instance, preferred angles from which an object may be retrieved or length of time it takes to retrieve up a particular object). The function to be maximized is

$$f(x) = f_1(\text{Value}) + f_2(\text{Accesses}) + \dots \quad (5)$$

subject to the dynamic constraints of the robot arm (simplifying from Equation (5)):

$$g(x) = \left(g_i(x) = [T_{Pickup}(x_i) + T_{Man}(x_i, x_{i-1})] - T_i \right) \quad (6)$$

Note that the access constraints are implicit in the set of *possible* scheduled points. All scheduled times outside of the feasible access time are assumed to be unscheduled points and are not considered in constraint calculations.

This optimization problem was run using a number of tools. Standard nonlinear programming algorithms[Mo92] were unsuccessful. Figure 2 shows a small field example run overnight using a nonlinear programming algorithm in which a human could easily eyeball a better schedule in a matter of seconds.

Our conclusion is that even though the problem (in simplified form) can be expressed mathematically, numerical solutions are not easy to find due to ill conditioning. Consequently a more robust and more flexible approach was explored: genetic algorithms.

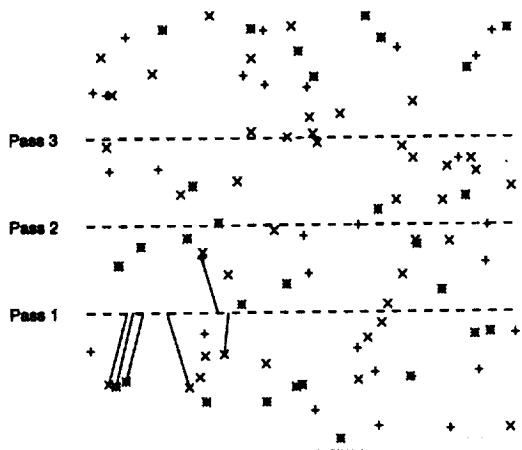


FIGURE 2. A disappointing result using non-linear programming

4.0 A Brief Introduction to Genetic Algorithms

The term *Genetic Algorithms* [Ho75] includes a broad class of iterative optimization techniques that employ methods that are modelled after the way evolution occurs by natural selection in biological systems. The traits common to all genetic algorithms are discussed in the following algorithm schema.

1. **Populations.** Instead of iterating on a single solution (as in most iterative optimization methods), a genetic algorithm begins with a set of (suboptimal) solutions. In keeping with the biological/evolutionary theme, the set of candidate solutions is called the *population*. The initial population may be arbitrarily or randomly chosen, or it may be given as an external input.
2. **Element transformation.** One or more elements are selected from the population, modified to produce a new, possibly better solution, and then put back into the population, replacing a then current population element. (See Figure 3.)

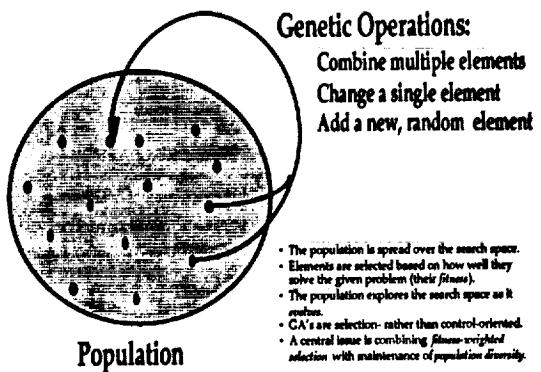


FIGURE 3. A generic genetic algorithm

A distinguishing feature of genetic algorithms is the manner in which solutions are modified and in some cases combined to produce new solutions.

- A solution may be modified to produce a new solution in a process called *mutation*. Nearly all optimization/search techniques use mutation in one form or another.
- Two or more solutions may be combined to produce a new solution. The process of combination can create new solutions that combine the best attributes of their predecessors in ways that are very unlikely under purely random stochastic methods. This is widely considered as one of the sources of the efficiency and broad applicability of genetic algorithms.
- 3. **Non-determinism.** Most search techniques typically explore a search space by applying transformations to known elements to produce other elements. This is true of genetic algorithms as well. Genetic algorithms differ from other search techniques in that they are able to take better advantage of the non-determinism present in many such transformations.

Since genetic algorithms work with populations of elements, if a transformation includes a non-deterministic feature, a genetic algorithm is often able to accommodate the multiple outcomes of that transformation. This is not to say that the population is allowed to grow exponentially. But since an element is not necessarily removed from a population after a transformation is applied to it, the same element may be transformed by a non-deterministic transformation multiple times, producing multiple different results. All of those results have a chance of entering the population. The better ones are more likely to stay in the population than the worse ones.

In practice, non-determinism means that search transformations often include probabilistic elements.

4. Fitness weighted selection. An application-specific evaluation function is applied to each member of the population to rank the solutions according to what is known as their *fitness*.

- When elements are selected for transformation, preference is given to selecting the higher ranking solutions.
- When an element is selected to be eliminated, preference is given to eliminating the lowest ranked solutions.

5. Iteration. The selection/transformation/replacement process repeats until some termination criterion is met. The best ranked solution(s) are then produced as output.

If genetic algorithms were no more than searches based on populations and non-determinism, they would differ little from a probabilistic variant of best-first search. As mentioned above, perhaps the most important feature of genetic algorithms is that they generate new population elements by combining existing population elements. The rationale behind this feature grows out of the observation that in many search problems, good solutions

often have features that are useful in many contexts.

The object retrieval problem provides some especially clear examples. If a large group of objects is close enough together to be retrieved by placing the net at a point that is within reach of all of them, that net placement is likely to be useful in many potential schedules. Similarly, if two groups of objects are sufficiently close that the second group can be retrieved by moving the arm minimally after retrieving the first group, that pair of arm placements will be a useful component in many schedules.

When a genetic algorithm allows two (or more) population elements to combine to produce a new population element, it allows useful features of the two elements to be combined in a single element.

Of course, when combining two elements, one does not necessarily know which features are useful. Even if one did, useful features are not always compatible. Non-determinism and fitness-weighted selection deal with that problem.

- Incompatible feature combinations produce poorly performing population elements, which are soon discarded.
- Useful features that are discovered independently generally survive in the population long enough to be combined in new population elements. For bit-string based genetic algorithms, such useful features are called *schemas*. Holland's Schema Theorem [Ho75] characterizes the transmission of useful schemas.

Genetic algorithms have been applied successfully to a wide range of optimization problems, such as the travelling salesman problem [Gr85], communication network design [Da87], natural gas pipeline control[Go83], image processing [Fi84], and other areas.

Genetic algorithms are a specific case of a more general concept called evolutionary com-

putation, which includes the related fields of artificial life, artificial evolution, and complex adaptive systems. Artificial life refers to simulations of agents acting in some simulated world. The agents are typically ranked according to their success in dealing with the simulated world, and genetic algorithms are used to evolve representations of successively better agents. The agents may interact individually with the simulated world, or the agents may interact with each other in cooperative or competitive ways within the simulation.

5.0 The Application of Genetic Algorithms to Object Retrieval

In our first attempt to apply genetic algorithms to the object retrieval problem we first used the general-purpose bit-string based genetic algorithm tool Genesis [Gr84] to optimize the numerical formulation described above. A satisfactory schedule for a single pass, shown in Figure 4, was generated with this method.

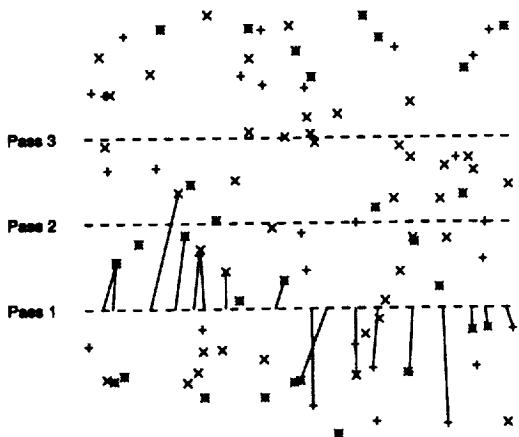


FIGURE 4. Result of applying a genetic algorithm applied to the numerical formulation

Compared to the results in Figure 2, the genetic algorithm shows marked improvement over

nonlinear programming. However, extension from the single pass case to a multi-pass case was not feasible due to the increase in the number of operations, computation time, and the complexity of adjudicating retrieving decisions. Thus a more problem-specific genetic algorithm was implemented. It includes a number of distinctive features.

1. The population consists of actual schedules, rather than bit strings. Significant care was taken to represent schedules in a way that was not only intuitive, but also space efficient and computationally efficient.
2. Since the population elements are schedules, the genetic operators are all problem-specific. (This is known as a *hybrid* genetic algorithm. [Da91]) Operators are defined that transform one or two existing schedules into a new schedule.
3. A number of innovative population management strategies were employed. As discussed above, genetic algorithms serve two masters: short term optimization (hill climbing) and diversity. On the one hand, it is desirable to climb whichever hill one is on; on the other, one doesn't want the entire population to be marooned on a suboptimal peak. New population diversity techniques were combined with greedy genetic operators as a way of achieving both objectives. This is discussed below.

5.1 Schedule representation

A schedule is a sequence of *appointments*, where an appointment is a robot x-position along with an arm (x, y) position. (Appointments are given in terms of robot x-positions instead of time since the robot moves at a constant rate.)

As an illustration, Table 1 displays the beginning of one pass of a schedule.

Except perhaps for the *Object windows* column, this table should be self-explanatory. The *Object windows* column shows each *object's*

window of accessibility in the current pass, i.e., the range of robot x-positions during which the object is directly accessible to the robot hand, i.e., without a net. If, because of the robot's minimum arm length, there is an internal subwindow during which the object is not accessible, the subwindow is shown in angle brackets. This happens to be the case for the first and third object in the first appointment but for none of the other objects in the table.

5.2 The selection/replacement cycle

As the earlier discussion of probabilistic search explained, genetic algorithms generally advance through a combination of exploration and combination. Both exploration and combination require that one take one or more elements from the search space and produce a new element *in the search space*. In many domains, robot object retrieval included, that is not a trivial task. The primary difficulty is that the operations that one performs on a search

space element do not always produce another valid search space element—in our case a schedule that satisfies the consistency constraints.

The primary consistency constraint on a schedule is that the robot be capable of moving its arm from one appointment to the next in the time allowed. A second consistency constraint is that no object be retrieved (or at least not be counted) twice.

Because it is not always easy to generate new population elements that satisfy the consistency constraints, production of new elements often involves two steps.

1. Generate a new element which may look like an element of the search space but which may or may not actually be a valid element of the search space.
2. Transform the new element into one that satisfies the consistency constraints.

TABLE 1. Example partial schedule (the first four appointments in one pass)

Robot x coordinate for this appointment	Robot net position for this appointment	Appointment value	Objects retrieved this appointment		
			Object positions	Object values	Object windows
0.000	(0.292, 0.232)	5	(0.273, 0.252) (0.275, 0.200) (0.319, 0.203)	1 2 2	[0.000 <0.185 - 0.361>0.569] [0.000 - 0.558] [0.035 <0.294 - 0.344> 0.603]
0.072	(0.090, 0.142)	8	(0.050, 0.135) (0.067, 0.130) (0.091, 0.142)	2 3 3	[0.000 - 0.301] [0.000 - 0.314] [0.000 - 0.346]
0.138	(0.247, 0.118)	7	(0.238, 0.126) (0.262, 0.150) (0.277, 0.092)	2 2 3	[0.000 - 0.482] [0.002 - 0.522] [0.061 - 0.493]
0.220	(0.257, 0.439)	6	(0.233, 0.471) (0.266, 0.426) (0.269, 0.439)	3 1 2	[0.000 - 0.480] [0.000 - 0.538] [0.004 - 0.535]
...

That is the process shown in Figure 5. The generation of a new element involves the following steps.

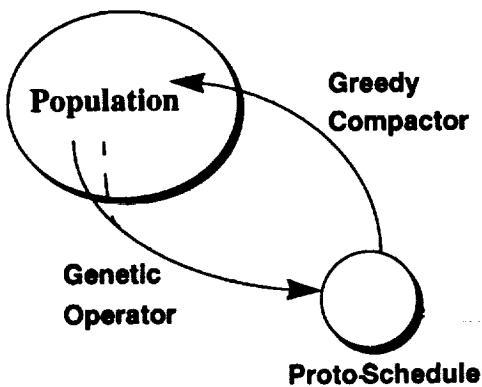


FIGURE 5. Population generation cycle

1. Select one or two elements from the population and transform it (or them) into what is called a proto-schedule, a structure that may or may not be a valid schedule.
2. Operate on the proto-schedule to produce a valid schedule.

The first transformation is the application of a genetic operator to one or two population elements. The second is the application of what we are calling a *schedule compactor*. The schedule compactor is itself a greedy scheduler. It transforms a proto-schedule, which may not satisfy the constraints, into an actual schedule, which does. We discuss the schedule compactor first and then the genetic operators.

5.3 The greedy schedule compactor

Like an actual schedule, a proto-schedule is a collection of passes, each of which is an ordered list of prospective appointments. A prospective appointment is a suggested arm position along with the list of the objects reachable from that arm position, as in Table 1. The only difference between a prospective appointment and an actual appointment is that a

prospective appointment does not have an assigned robot x-position. The compactor's job is to associate robot x-positions with the given prospective appointments. It takes a table such as Table 1, but with no information in the first column, and for each prospective appointment it either assigns a robot x-position or deletes the appointment.

The schedule compactor does that job greedily. For each pass in the proto-schedule, the compactor makes a single traversal of the list of prospective appointments and throws out the ones that are incompatible with the constraints: the robot arm length, the time required to complete the previous appointment, and the maneuver time required to move the arm from the previous appointment. The compactor is greedy in the following ways.

1. Prospective appointments that are consistent with the constraints are scheduled as early as possible, i.e., the smallest possible robot x-coordinate consistent with the indicated net placement is selected.
2. Whenever possible, the compactor shifts objects from one appointment to an earlier appointment. If this involves shifting the arm position of the earlier appointment, that is acceptable as long as no objects are lost.
3. Whenever possible, the compactor adjusts an appointment's x-y arm position slightly if doing so would enable the robot to retrieve additional objects.

In addition, the compactor drops from prospective appointments any objects that have already been scheduled. Prospective appointments that are left without any objects are dropped entirely.

The compactor has a probabilistic feature built into it. The compactor must compact all passes of a proto-schedule. The order in which those passes are compacted may make a difference. If an object is reachable during multiple passes (as many are), the pass during which it is retrieved may affect other objects. To allow for

all possibilities, the greedy compactor first constructs a random permutation of the passes and then compacts them in that order.

5.4 Genetic operators

The genetic operators perform two functions.

1. They are used to explore the search space. Typically, these are the so called *mutation* operators. A mutation operator takes a population element, i.e., a schedule, and transforms it into a proto-schedule. These transformations may or may not actually improve the schedule. They are simply exploration steps.
2. They combine two schedules to produce a new proto-schedule. The primary function of the combination operators is to combine features of good schedules in the hope of producing a better schedule.

The following mutation operators are defined. Most of them have a great many opportunities for non-determinism. These operators may or may not produce valid schedules. If they don't, the compactor makes the needed repairs.

- **change the pass of an appointment.** Move an appointment from one pass to another.
- **schedule an unscheduled object.** Retrieve an object that is not currently in the schedule and create an appointment for it.
- **interlard some unscheduled objects.** Sort a random selection of the unscheduled objects; allocate them to passes in which they have windows; and merge them with the current schedule. The merge process is the same as that explained below under **merge two schedules**. This operation is similar to **schedule an unscheduled object**. The difference is that it attempts to schedule collections of unscheduled objects instead of just one.
- **schedule a group.** Select all the unscheduled objects in a group and schedule appointments for them. (Recall that a group

is an all-or-nothing affair. The robot does not get credit for retrieving objects in a group unless the entire group is retrieved.)

There is no corresponding unschedule-group operation. Instead, whenever an element of the population is selected for transformation, one of the groups is (probabilistically) unscheduled.

- **exchange appointments.** The order of two adjacent appointments is switched.
- **generate a random schedule.** Generate a new, random proto-schedule. There are a number of probabilistic elements involved. The objects may first be ordered by value. In addition, the proto-schedule is generated by sorting the objects (one object per appointment) according to either x-y position or start-of-window-in-pass. If an object may be retrieved in a number of passes, the pass to which it is assigned is also deterministically probabilistically.

There is a single combination operator.

- **merge two schedules.** This operator combines and compacts two population elements. Appointments from corresponding passes of two schedules are merged, greedily. The merged result is guaranteed to conform to the constraints.

The order condition that drives the merge is a combination of appointment x-position and appointment value. If the first available appointment from one schedule is both earlier than and more valuable than the first available appointment from the other schedule, it is selected. Otherwise, the schedule from which the next appointment is taken is selected at random.

6.0 Sustainable Genetic Algorithms

In the actual application, we sometimes want to run the genetic algorithm for an extended

period. On other occasions, we need a reasonably good answer after only a relatively short run. We therefore want a genetic algorithm that can both (a) provide good results relatively quickly and (b) continue to improve if left to run for an extended time. We call a genetic algorithm with the second property *sustainable*.

One can produce reasonably good results quickly by including among our genetic operators, heuristics defined for the scheduling problem. Unchecked, however, this practice leads to population convergence at local maxima. Special techniques must be made to avoid such convergence. The following first discusses the mechanisms underlying population convergence and then describes ways to combat it.

6.1 Mechanisms Underlying Population Convergence

The essence of a genetic algorithm is probabilistic exploration of a (typically very large) search space. This exploration proceeds as a population of search space elements traverses the search space. Two main drivers propel the population through the space.

- Through mutation, subpopulations tend to climb search space hills.
- Through element combination (most commonly via the genetic operator called “crossover”), the population tends to accumulate in regions of the search space that represent solution features that can be combined in useful ways.

In GAs, these regions of the search space containing elements that all include some specified feature are called *hyperplanes*. This term derives from the origins of GAs in which population elements were represented as bit strings. Under that representation, a hyperplane is the set of search space elements that have some fixed values in specified bit positions. In the case of non-bit string GAs, the notion of a hyperplane may be generalized to the

set of search space elements that include a particular solution feature.

Traditionally, solution features that define hyperplanes have been called *schemas*. In scheduling, a schema would typically be a sequence of scheduled events, i.e., a schedule fragment. All search space elements that contain a particular schedule fragment may be considered to be on the same hyperplane. (Each search space element, i.e., a complete schedule, may lie on many intersecting hyperplanes simultaneously.) Useful schedule fragments will tend to be retained in the population. Hence, the population will tend to accumulate on the hyperplanes defined by useful features.

Since population size generally stays relatively constant over extended periods, and since elements are retained in the population based on how well they solve the given problem, the overall quality of the population tends to increase. This means that subpopulations will tend to accumulate both on local maxima (hills) and on useful hyperplanes.

As the overall level of the population increases, some of the local maxima will be submerged beneath the median population level and will vanish from the population. Similarly, some of the hyperplanes will no longer include any elements that are competitive, and they too will disappear. Eventually, these effects can cause the population to converge to a small set of related solutions.

6.2 Combatting Population Convergence

If population improvement and convergence is the first principle of genetic algorithms, a second (and countervailing) principle is the need to maintain population diversity. Premature convergence to a suboptimal solution is exactly what genetic algorithms are intended to avoid.

To counter the genetic algorithm tendency toward convergence, mechanisms may be imple-

mented to force the population to maintain diversity. The most common of these is simply to restart the GA from scratch after it has converged on a solution. One can run the GA independently a number of times and select the best result.

Other diversity-maintaining mechanisms include:

- a) **Mutation.** Mutation serves two purposes. Mutation of a population element in minor ways drives hill climbing. Mutation of a population element in major ways introduces new areas of the search space into the population. The most extreme form of mutation is the introduction of elements selected at random from the search space.
- b) **Dynamic changes to the algorithm that determines which population elements will be selected for reproduction and which will be discarded.** The more the replacement algorithm favors the better population elements the more it tends to drive the population toward convergence. Modification of the selectivity of the selection algorithm can be used to encourage diversity.
- c) **Multi-modal evaluation functions.** A GA with multiple objectives will not converge to a single solution.
- d) **Evaluation functions that penalize population elements that are “too close together.”** If one has a metric on the search space, and if one discourages the population from gathering in a “small area” of it, diversity can be encouraged.
- e) **Embedding the GA population in a data structure that artificially creates a distance between elements.** It is often difficult to define a useful measure of the distance between two population elements. An alternative is to embed the population in a space that has an easy distance metric. The most common of these structures is a torus-shaped grid. Elements are assigned

positions on the grid. Those close to each other are permitted to interact more frequently than those distant from each other. This technique spreads out the population so that a high valued population element in one part of the landscape does not immediately invade the entire population.

- f) **Running the GA with multiple subpopulations that are partially isolated from each other.** (This can be thought of as a variant of (e).) Instead of a single population, one runs multiple GAs in which each population is allowed to evolve more or less independently. Occasionally, elements are permitted to migrate from one population to another.

It has been widely observed that almost all genetic algorithms are amenable to parallel processing, either on a tightly coupled parallel computer, or on a collection of loosely coupled workstations communicating over a network. In particular, variant (f) offers an ideal avenue for parallel implementation with minimal overhead due to inter-process communication.

6.3 Our Approach in the Object Retrieval Problem

We achieve sustainability through the use of two techniques: continual injection of new, random elements and tournament selection with varying competition levels, i.e., methods (a) and (b) above.

Continual injection of new, random elements is just as it sounds. Instead of transforming an existing schedule, an entirely new schedule is generated.

Our technique of varying competition levels is based on our use of a steady-state population. There are two standard approaches to population management: generational and steady state. We follow the steady state approach. Within the context of a steady-state population, the central issue is the selection of (a) the

elements to be transformed and (b) the elements to be discarded.

We use a variant of tournament selection to make both selections. To select an element for transformation, a subset of the population, *the selection pool*, is chosen randomly and uniformly from the entire population. The best (or best two) element(s) of that pool are selected for transformation. To select an element to be discarded, we again choose a subset of the population; the worst element of the selection pool is selected for deletion.

Since elements are included in the selection pool with equal probability, the size of the selection pool is *inversely related* to the selectivity of the search. If the pool size were 1, one would be selecting (for transformation or deletion) an element uniformly from the population, i.e., with no regard for how well the element solved the problem. This would minimize convergence, but it would also minimize the likelihood that good features would be exploited.

On the other hand, were the pool to be the entire population, one would always select the best element(s) for transformation and the worst for deletion. This would maximize convergence, but it would virtually eliminate significant diversity.

Our strategy is to allow the size of the selection pool to vary. Whenever a new best element is produced, we reduce the level of competition by shrinking the pool size. This increases the probability that an element included in the pool will actually be selected. The effect is that the population to spreads out over the search space. As elements are generated with no new better solution produced, we gradually tighten the competition by enlarging the selection pool size. This focuses the search among the better elements of the population.

The overall effect is similar to the annual cycle of championships in competitive sports. At the beginning of the season, competition is spread

out. New entrants have an opportunity to be seen. This is comparable to local tournaments. As the season progresses, competition tightens; only the better entrants remain in the field. (Unlike sports, our population does not shrink, but the likelihood decreases that a poorly performing element will be selected for transformation.) Toward the end of the season, the selection pool is large and competition is extreme. Only "world class" elements survive. But as in competitive sports, because the entire process is probabilistic, there is always a chance that an underdog can make it to the "finals."

This seasonal cycle repeats itself continually. The selection pool size starts low, grows slowly, and then restarts at a low value for the next season. New elements with innovative features continually arise to challenge and add value to the current champions.

7.0 Results

The following plots illustrate the results of the genetic algorithm. For simplicity and consistency, these plots are based on a run with the following parameters.

Object field. The testbed example included 100 objects with values of 1, 2, and 3. The total of all objects (and hence the best possible schedule) was 201.

Groups. There were 5 all-or-nothing groups. The largest had 11 objects with a total value of 22; the smallest had 5 objects. No object was in more than one group. Of the 100 objects, 41 were in some group.

Robot constraints. The robot arm was required to be nearly perpendicular to the direction of travel, at least 75 degrees. The robot had a net of radius 0.04 at the end of the arm. For each appointment, the robot was required to keep the arm end point stationary while it traversed a horizontal distance of 0.05. The robot was able to maneuver the arm (in a straight

line from point to point) at 10 times the rate of its horizontal motion. The arm is assumed to move at a constant rate; there are no start-up or terminate arm motion penalties.

In this run, which showed typical results, approximately 40,000 schedules were considered. The best schedule had a value of 162. During the run, nearly 6000 random schedules were generated. The best of these had a value of 100. We take this as confirmation that the genetic operators added value to the search.

Figure 6 is a "fishbone" diagram. It shows the object field with the passes drawn as horizontal dashed lines. Each appointment is shown as a shadowed circle. The robot arm is shown as a line connecting the appointment to the position of the robot at the time of the appointment.

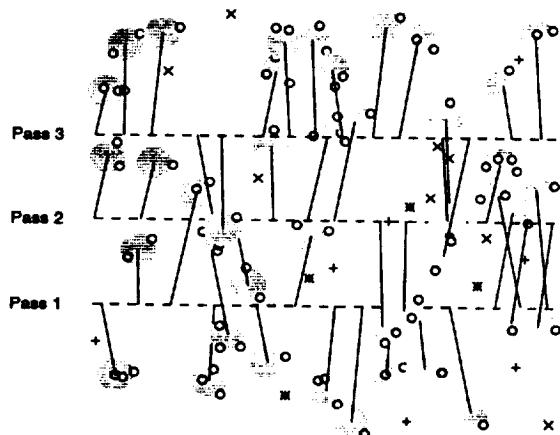


FIGURE 6. Arm positions

Retrieved objects are shown as O's; unretrieved objects are shown as +'s, X's and *'s. Recall that some of the objects are in all-or-nothing groups. Four of the five groups were retrieved in their entirety. The largest group was not retrieved. Of the 11 objects in it, 3 were retrieved anyway even though they contributed no value to the schedule. Objects that appear to be easy picking but were not retrieved belong to the unretrieved group.

Figure 7 shows the trajectory of the end of the arm for this schedule. As this figure makes ap-

parent, no effort was made to minimize arm motion. The only criterion for preferring one schedule over another was the total value of objects retrieved. If desired, such additional criteria could be added easily.

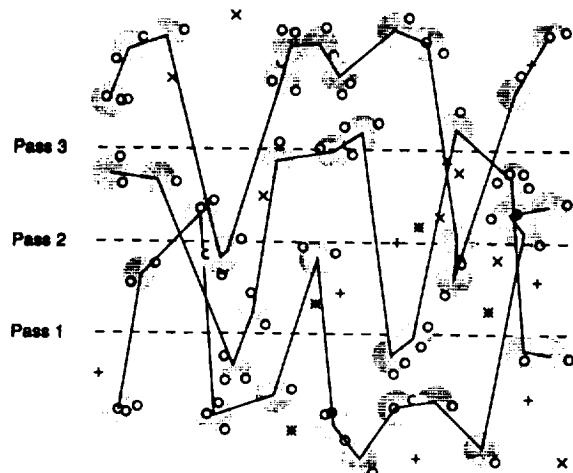


FIGURE 7. Arm trajectory

As a contrast, Figure 8 shows a schedule generated by a robot without a net: the "hand" must retrieve each object individually. Not surprisingly, this schedule is less effective in retrieving objects. More importantly, generating schedule variations of this sort turns out to be quite simple once the hybrid genetic algorithm framework is in place.

For each schedule, a partial family history is maintained. In particular, the values of the schedule's parent (or parents) is kept along with the history of the better parent. This associates with each schedule a record of that schedule's best parent back to the time when it was created as a random schedule.

Figure 9 compares the family history of the eventual Best Schedule with the best schedule in the population (Best of Population) at the time. This plot illustrates two features of this run.

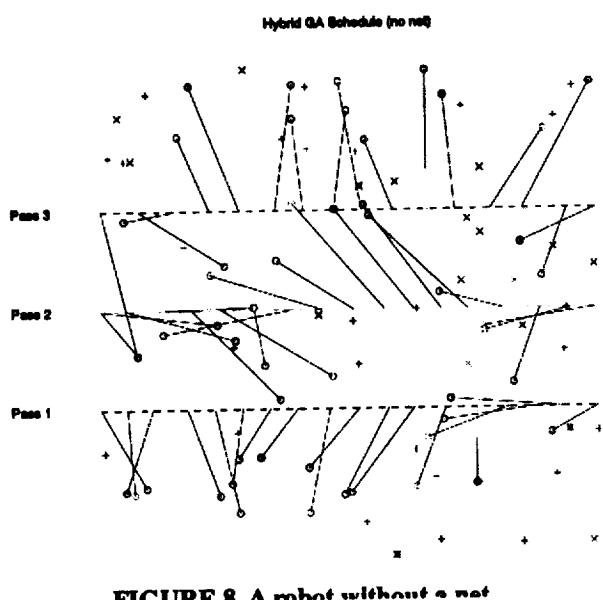


FIGURE 8. A chart without a net

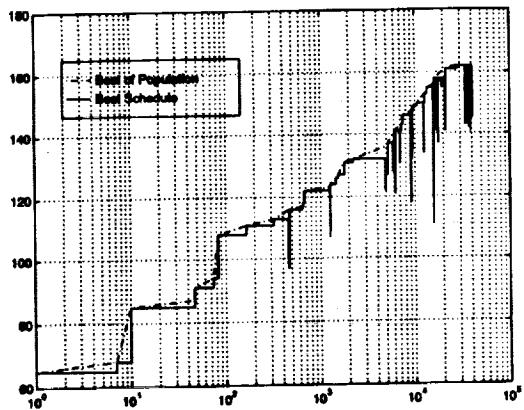


FIGURE 9. Best schedule genealogy

1. The eventual best schedule was close to the best during its entire genealogical history. (Had it not been, it would probably have gone extinct.)
2. Significant schedule degradations occurred before many of the advances. (Unfortunately, information regarding the nature of the degradations and whether or not they were prerequisite to the subsequent advances is not available.)

8.0 Future Work

Currently we are continuing to explore GA-based scheduling in two primary areas: scalability studies and new genetic operators. We are also looking for additional applications of GA-based scheduling/optimization.

In order to understand the scalability of the hybrid GA-based scheduler, we are currently performing a series of timing/profiling experiments over a wide range of problem sizes. By independently varying the size of the field and the number of objects, these scheduling runs will enable us to determine the dependence of the scheduler on the number and density of objects vs. the length of the resulting schedules for each pass. We are currently using a simple termination condition for the experiments: the algorithm stops after 10,000 schedules have been generated and evaluated. Preliminary results show a roughly linear increase in runtime vs. problem size for most cases.

Since the **merge two schedules** operation is one of the most time consuming hybrid genetic operations, it is natural to consider a more traditional genetic crossover operator. This new genetic operator would combine two schedules to produce a new population element according to the following process. The two schedules, represented as lists, will first be aligned with respect to time (in some way still to be determined,) and then the two lists will be traversed in lockstep with respect to time. As the two lists are traversed, the genetic operator copies from them to create the child list, with random crossovers from one parent to the other. Finally, the resulting proto-schedule is processed through the compactor to produce the new population element. It will be interesting to observe the effect that this new operator has on computational efficiency, population diversity, and the resulting schedules.

9.0 Conclusions

This study has demonstrated the applicability of hybrid genetic algorithms to a difficult scheduling problem. This problem resisted solution using more traditional techniques. Yet with a hybrid genetic algorithm good schedules have been generated.

Hybrid genetic algorithms differ from traditional genetic algorithms in that they make use of knowledge representation strategies and heuristics from the problem domain. This was very important for this problem in two ways.

1. It enabled us to represent the population of schedules in an efficient manner, precluding the need to transform bitstrings back and forth to schedules.
2. It allowed us to apply heuristics from the scheduling domain. Without these heuristics it is unlikely that we would have been able to solve the problem.

The primary genetic algorithm challenge is population management: how to manage the population so that (a) no one subpopulation drives out all others and (b) the various domain heuristics all have a reasonable opportunity to be applied. Our strategies of continually introducing new random elements and varying the competition level appear to have achieved these objectives.

The genetic algorithm paradigm provides a framework for selection-based search. As such it avoids many of the problems inherent in control-based search strategies. The inclusion as genetic operators of domain-specific heuristics from control-based algorithms allows one to combine the best of both approaches.

Acknowledgment

This work was performed at the Aerospace Corporation under sponsorship of its Aerospace Sponsored Research Program.

References

- [Al90]. Allen, J, J Hendler, A Tate, (eds.), *Readings in Planning*, Morgan Kaufmann, San Mateo, Ca. 1990.
- [Da87]. Davis, L and S Coombs, "Genetic algorithms and communication link speed design: theoretical considerations," *Proceedings of the Second International Conference on Genetic Algorithms*, pp 252-256, 1987.
- [Da91]. Davis, L (Ed), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.
- [Fi84]. Fitzpatrick, JM, JJ Grefenstette, and D Van Gucht, "Image registration by genetic search," *Proceedings of the IEEE Southeast Conference*, pp 460-464, 1984.
- [Go83]. Goldberg, DE, *Computer-Aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning*, Doctoral Dissertation, University of Michigan, 1983.
- [Gr84]. Grefenstette, JJ, "GENESIS: a system for using genetic search procedures," *Proceedings of a Conference on Intelligent Systems and Machines*, pp 161-165, Rochester, MI, 1984.
- [Gr85]. Grefenstette, JJ, R Gopal, B Rosamita, DV Gucht, "Genetic algorithms for the traveling salesman problem," *Proceedings of International Conference on Genetic Algorithms and their Application*, pp 160-165, Lawrence Erlbaum & Assoc., NJ 1985.
- [Ho75]. Holland, JH, *Adaption in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, Michigan, 1975.
- [Mo92]. Moler, C, J Little, and S Bangert, *PRO_MATLAB for Sun Workstations: User's Guide*, The MathWorks, Sherborn, Ma 1993.



A Rule-Based Shell to Hierarchically Organize HST Observations

Ashim Bose and Andrew Gerb

bose@stsci.edu and gerb@stsci.edu
Space Telescope Science Institute
Baltimore, MD 21218

SG-82

47259

P. 12

1. Abstract

An observing program on the Hubble Space Telescope (HST) is described in terms of exposures that are obtained by one or more of the instruments onboard the HST. These exposures are organized into a hierarchy of structures for purposes of efficient scheduling of observations. The process by which exposures get organized into the higher-level structures is called *merging*. This process relies on rules to determine which observations can be "merged" into the same higher level structure, and which cannot.

The TRANSformation expert system converts proposals for astronomical observations with HST into detailed observing plans. The conversion process includes the task of merging. Within TRANS, we have implemented a declarative shell to facilitate merging. This shell offers the following features: a) an easy way of specifying rules on when to merge and when not to merge, b) a straightforward priority mechanism for resolving conflicts among rules, c) an explanation facility for recording the merging history, d) a report generating mechanism to help users understand the reasons for merging, and e) a self-documenting mechanism that documents all the merging rules that have been defined in the shell, ordered by priority.

The merging shell is implemented using an object-oriented paradigm in CLOS. It has been a part of operational TRANS (after extensive testing) since July 1993. It has fulfilled all perfor-

mance expectations, and has considerably simplified the process of implementing new or changed requirements for merging. The users are pleased with its report-generating and self-documenting features.

2. Introduction

2.1. Planning and Scheduling HST Observations

Once a proposal for observing with the HST has been approved, the astronomer submits a detailed observing plan. This plan contains specific exposures, instrument configurations, and constraints on exposures. There are a variety of scientific reasons why an astronomer might place additional constraints on exposures and between exposures. For example, exposures may be designated as acquisition or calibration exposures. Some exposures might be executed at particular times, specific orientations on the sky, or within a designated time interval. In the case of time-variable phenomena (e.g. binary stars, Cepheid variable stars) the proposer may require repeated observations at specific time intervals (Miller and Johnston 1991).

2.2. TRANSformation - The Big Picture

The process of converting a proposer's specification into a form suitable for scheduling is called *transformation*. Transformation involves several tasks including determining the ordering of the observations, grouping to minimize telescope

movement, instrument reconfiguration and other overheads, providing extra observations and instrument activities necessary to obtain the requested data, and organizing observations into a hierarchy of higher-level structures for purposes of scheduling observations.

The TRANSformation expert system (TRANS) converts proposals for astronomical observations with the HST into detailed observing plans. In other words it performs all the tasks associated with the process of transformation, as described in the previous paragraph. For a detailed description of its workings, see (Gerb 1991). An extension of Common LISP (Steele 1990) was used for its implementation.

3. "Merging" Observations into a Hierarchy

Merging is defined as the process of organizing observations into a four-level hierarchy for purposes of efficient scheduling (Fig. 1).

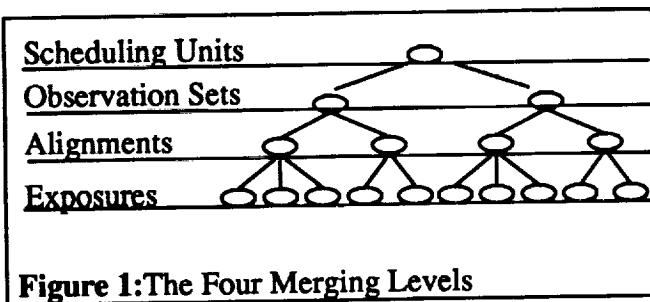


Figure 1:The Four Merging Levels

At the lowest level are the exposures that are obtained from the observing plans. During the process of merging, the ordered exposures are grouped into contiguous disjoint sets called alignments. Alignments are then grouped into observations sets (obsets) and finally, obsets are grouped into scheduling units.

All exposures in an alignment must use the same HST pointing and orientation, must generate science and engineering data at the same rate, and must have only small time gaps between successive exposure members. Grouping of exposures into alignments is important for two reasons. First, exposures in the same alignment can be commanded for much more efficient use of spacecraft time. Second, alignments are the basic

units of planning for the downstream scheduling system.

Obsets are groups of alignments that can be executed without a change in the operating mode of the pointing control system. HST usually depends on positional monitoring of pairs of stars (called guide stars) to maintain its pointing. A series of alignments can be in the same obset if they all can use the same guide star pair, or if they do not use guide stars.

Scheduling Units are groups of obsets that are scheduled together. When scheduling an obset requires the next obset to be scheduled immediately afterwards, both are placed in the same Scheduling Unit.

While merging, the ordering of objects is preserved, i.e. there is no change in the ordering of observations due to merging. So, if there are n exposures in an observing plan, numbered 1 thru n , and they are in ascending order, we first create a new alignment (numbered 1) for exposure 1. Then we see if exposure 2 can be put in alignment 1:

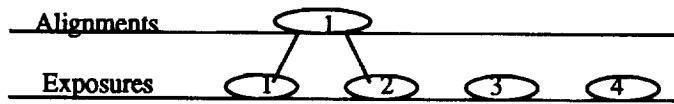
- If yes, exposure 2 gets included into alignment 1. Now, we see if exposure 3 can be included in alignment 1. If yes, alignment 1 now has three exposures. If no, we create a new alignment (numbered 2), and exposure 2 goes into alignment 2.
- If no, alignment 1 has exposure 1 and we create a new alignment (numbered 2) that has exposure 2. Now, we see if exposure 3 can be put in alignment 2. If yes, alignment 2 now has two exposures, exposures 2 and 3. If no, exposure 3 goes into a new alignment (numbered 3).

We repeat this process for exposures 4 until n , always considering the latest new alignment formed, for exposure inclusion. Once we have all the alignments, we step up a level, and go through the alignments in order, grouping them into obsets. Once this is done, we group the obsets into scheduling-units. The exposure->alignment merging process is also illustrated in Fig. 2

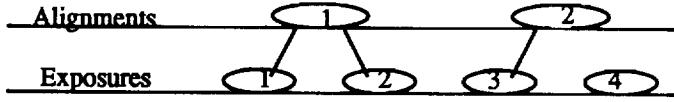
An unmerged observing plan



Exposures 1 and 2 can be merged into the same alignment, because of a merging rule



Exposure 3 starts a new alignment due to a breaking rule



Exposure 4 can be merged into alignment 2 because of a merging rule

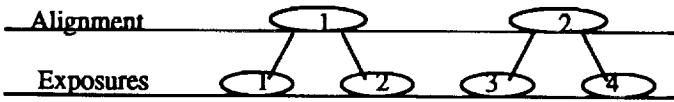


Figure 2:An illustration of the process of merging at the exposure->alignment level

Within TRANS, merging can be achieved in two ways. Under *manual-merging*, the user prescribes to TRANS how an observing plan should be merged. This is done through a “merging-file” that the user sets up before running TRANS on an observing plan. Under *automatic-merging*, the entire merging process is left to the software. Decisions on whether to merge an object into a higher-level-object are based on rules. Rules are of two types: merging, or breaking. A rule of type merging (henceforth referred to as a *merging rule*) specifies a set of conditions under which a lower level object can be included in a higher level object (that may already contain other lower level objects). A rule of type breaking (henceforth referred to as a *breaking rule*) specifies a set of conditions under which a lower-level-object cannot be included in a higher-level-object that already contains other lower-level-objects. Conflicts might arise when both merging and breaking rules may be applicable. So, a scheme for *conflict resolution* is important.

The following requirements were defined for the implementation of the TRANS merging shell:

- an easy way of specifying merging and breaking rules
- a priority mechanism for resolving conflicts among rules
- an explanation facility to document the merging history for an observing plan under automatic merging
- a facility to validate an observing plan that is being manually merged
- a self-documenting mechanism that documents all the merging rules that have been defined in the shell, ordered by priority.

4. A Declarative Merging Shell

Automatic merging, as described in the previous section is implemented using the TRANS Merging Shell. This shell provides mechanisms for encoding the rules, and defining and using any associated data structures. It also provides explanation and self-documenting facilities. In describing the shell, a notation very close to LISP (and

CLOS) syntax will be used because of the need to include implementation details, where necessary.

4.1. The Mechanisms for Declaring Knowledge

An object-oriented approach is taken to implement the merging shell. There are two kinds of objects: *data* objects, and *declarative* objects. The data objects are of the following type: exposure, alignment, obset, and scheduling-unit. These objects are doubly-linked to preserve ordering and facilitate object-traversal in both directions. The declarative objects are the rules and the slots. The rules are used to encode the merging criteria. The slot objects are used to encode knowledge used to populate some of the slots in the data objects.

Rules are the primary means of encoding the criteria for merging. A rule is defined using the construct *define-rule*. Each define-rule declaration results in the creation a rule object. The template for a rule is described in Fig. 3.

```
(define-rule
  :type <rule-type>
  :name <rule-name>
  :level <rule-levels>
  :instrument <rule-instruments>
  :priority <rule-priority>
  :test <rule-test>
  :description <rule-description>
)
```

Figure 3:The Rule Template

A description of the various parameters in the rule template is in order.

1.<*rule-type*> can be either :merging or :breaking.

2.<*rule-name*> is a short descriptive string that succinctly conveys the meaning of the

rule. It is used for identifying the rule in the explanation and self-documenting mechanisms, and so has to be unique.

3.<*rule-levels*> are the one or more merging levels the rule is applicable at, i.e. 'ex->al, or 'al->ob, or 'ob->su. The keyword :all may be used if the rule is applicable to all merging levels.

4.<*rule-instruments*> are the one or more instruments the rule applies to. Again, the keyword :all may be used if the rule is applicable to all instruments.

5.<*rule-priority*> is a real number. It establishes the priority of a rule, to aid in conflict resolution. The larger the number, the higher the priority of the rule. While resolving conflicts, a rule with higher priority takes precedence over a rule with lower priority.

6.<*rule-test*> is the symbolic expression that determines if the rule should fire. This expression is encoded in LISP. References to the current higher-level-object under consideration (*self*), and lower-level-object under consideration (*obj*), can be made in the symbolic expression.

7.<*description*> is a string that contains the detailed description of the rule in english. It is used by the self-documenting mechanism.

Note that the <*rule-test*> corresponds to the antecedent part of a traditional production rule. The symbolic expression that is <*rule-test*> can be arbitrarily complex and can refer to any of the properties of the lower level objects being considered for merging. It is evaluated by the inference engine. If the result is a non-null value, the rule is considered to have fired or activated. The role of consequent is played by the <*rule-type*>, which indicates the action to take in case the <*rule-test*> is "true".

An example of a rule is shown in Fig. 4.

```

(define-rule
  :name "BREAK EXPOSURES THAT
         DO NOT HAVE IDENTICAL
         ORIENTATION"
  :type :breaking
  :level :all
  :instrument :all
  :priority 2
  :test '(not
    (identical-orientation-p
      (first-ex-in-self self)
      (first-ex-in-obj obj))
  )
  :description
  "Break an exposure into a new SU if
  it does not have identical upper
  and lower limits for absolute and
  nominal orientations as the
  exposures in the previous SU."
)

```

Figure 4:An Example of a Rule Declaration

This rule enforces the condition that observations that do not have identical spacecraft orientations*, should not be grouped into the same higher level object. Hence, this rule is of type “breaking”, and applies to all merging levels (ex->al, al->ob, and ob->su). It is applicable to all instruments, and has a low priority. It activates when the “identical-orientation-p” test fails. This test is performed using the first lower-level object in the latest higher-level object (denoted by

*For a detailed description of how orientation constraints are dealt with in TRANS, see (Bose and Gerb 1994).

“self”) and the current lower-level object under consideration (denoted by “obj”).[†] So, if we are at the lowest level of merging (exposure->alignment), and the latest alignment to be created is alignment 3, with exposures 7, 8, and 9, and the object under consideration is exposure 10, we merge exposure 10 into alignment 3 only if it has the identical orientation as exposure 7, otherwise exposure 10, starts a new alignment (alignment 4). Considering another case, let us assume we are merging at the intermediate level (alignment->obset), and the latest obset to be created is obset 2, with alignments 2 and 3, and the alignment under consideration is alignment 4. Further assume alignment 2 has exposure 3 as its first exposure, and alignment 4 has exposure 10 as its first exposure. If the orientations of exposures 3 and 10 are identical, alignment 4 gets merged into obset 2, if not it starts a new obset (obset 3).

The properties of a higher-level-object keep changing as new lower-level-objects are included within it by the inferencing mechanism. These properties are defined using the construct *define-slot* (Fig. 5).

```

(define-slot
  :object <object-types>
  :name <slot-name>
  :initialize-with <initial-value>
  :update-with <update-expression>
  :update-after <other-slots>
)

```

Figure 5:The Slot Template

A description of the various parameters of the slot template is next.

[†]The use of the word “self” to refer to the current higher-level-object is not without significance. A rule can be considered a method for the higher-level-objects for the merging level of a rule. “Self” would then refer to the current higher-level-object for which the rule was being executed. For slot updates, “self” appears to be a good choice for obvious reasons.

1. *<object-type>* is a symbol representing alignment, obset, or scheduling-unit.
2. *<slot-name>* is a symbol that serves as a unique identifier for this slot.
3. *<initial-value>* is the value with which the slot in the relevant data object is initialized when the object is created.
4. *<update-expression>* is a symbolic expression that, when evaluated, yields the value associated with the slot. This expression can be arbitrarily complex, and can include references to other objects.
5. *<other-slots>* are the slots that should be evaluated before this one. This feature enables an ordering in the evaluation of the slots.

Both rule and slot declarations result in the creation of objects of the corresponding types. These are in addition to the exposure, alignment, obset, and scheduling-unit objects (data objects) that are created as needed. Note that slot objects contain information on attributes of the data objects. Since the *<update-expression>* may contain a reference to another slot in the same object, it is important to specify the slot-dependencies through *<other-slots>*.

An example of a slot declaration is shown in Fig. 6.

```
(define-slot
  :name 'primary-priority
  :object 'alignment
  :initialize-with -1
  :update-after 'primary-exposure
  :update-with
    '(cond
      ((equalp obj (primary-exposure self))
       (primary-exposure-priority obj))
      (t (primary-priority self)))
    )
)
```

Figure 6:An Example of a Slot Declaration

This slot is defined for data objects of type alignment. Whenever an alignment object is created, a slot called “primary-priority” is automatically created and initialized to -1 for the alignment. Whenever a new lower level exposure object is added to the alignment object, all the slots in the alignment object are updated with the result of the evaluation of the “update-with” expression. In this case, since the update-with expression contains a reference to another slot called “primary-exposure”, the “primary-exposure” slot needs to be populated before the “primary-priority” slot.

4.2. The Inference Engine

The Inference Engine (IE) uses the knowledge encoded in the declared objects alongwith the data in the data objects to accomplish the process of merging. The algorithm used is shown in pseudo-english in Fig. 7.

Algorithm Merge:

```
From lowest to the highest merging level {
  set obj to first lower-level-object ;
  while lower-level-objects remaining {
    deduce decision based on self and obj
    if decision is to merge {
      merge obj into self
      update slots in self }
    else
      set self to new higher-level-object;
      set obj to next lower-level-object
  }
}
```

Figure 7:Algorithm Merge Used in the Inference Engine

Merging commences at the lowest (exposure->alignment level), and then proceeds to the alignment->obset level, and finally obset->scheduling-unit level. The lower-level objects are merged into the higher-level objects in order, at each level. The decision to merge a lower-level-object into a higher-level-object is based on the priority of the rule that was activated. Rule activation is

based on the result of the evaluation of the $\langle rule-test \rangle$, given the current values of *slot* and *obj*. Algorithm deduce-decision which performs rule-activation is shown in Fig. 8.

Algorithm Deduce-Decision:

```

set decision to "break";
set current-priority to highest rule-priority;
while no rule has been activated and there
are rules remaining {
    if all current-priority rules have been
        exhausted
        set current-priority to
            next lower rule-priority;
    set rule to next unconsidered rule with
        current priority;
    evaluate  $\langle rule-test \rangle$  using self and obj;
    if result is "true" {
        ;rule has been activated
        record  $\langle rule-name \rangle$ ;
        if  $\langle rule-type \rangle$  is "merging"
            set decision to "merge"
    }
}
return decision
}

```

Figure 8:Algorithm Deduce-Decision Activates Rules

It operates by attempting to activate rules in descending order of priority. As soon as a rule activates, it returns the decision based on the type of the rule. The restriction that rules with the same priority have to be of the same type simplifies rule-ordering (before activation) and conflict-resolution (after activation). Rule-ordering for rules with the same priority is no longer important because all rules have the same consequence, i.e. merging or breaking. Conflict-resolution for rules with the same priority does not arise, again because all rules have the same consequence, and so there are no conflicts. If no rules are activated,

the default decision is "not to merge", or to "break".

5. Examples of Merging Shell Usage

Merging shell usage will be illustrated with the help of two examples.

5.1. Example 1

In this example, we make slot and rule declarations to ensure that observations that use the Faint Guidance Sensors (FGS) should be grouped into different obsets if they have different spacecraft pointings. In order to implement this requirement at the alignment->obset merging level, and keeping in mind that an alignment may have several exposures, we make use of the concept of primary-exposure within an alignment, which determines the pointing of the alignment. We declare a slot called primary-fgs for obsets, that should contain a reference to the first primary exposure within it if the exposure happens to be an FGS observation (Fig. 9).

```

(define-slot
  :name 'primary-fgs
  :object 'obset
  :initialize-with nil
  :update-with
    '(or (primary-fgs self)
      (let ((ex (first-ex obj))
            (pri (primary-exp obj))
            (fgs nil)))
        (while (and (not fgs) ex)
          (when
            (equalp (si-used ex) )
            (set fgs t))
          (setq ex (next-ex-in-al ex)))
        (when fgs pri)
      )
    )
)

```

Figure 9:A Slot Definition for Primary-FGS

The rule declaration is shown in Fig. 10. The *<rule-test>* essentially states that if *self* contains a primary FGS exposure, and *obj* also has a primary FGS exposure, and the two primaries do not have the same pointing, then *obj* should start a new obset.

```

(define-rule
  :name "FGS ALIGNMENTS WITH
         DIFFERENT POINTINGS"
  :type :breaking
  :level 'al->ob
  :priority 4
  :test '(let
    ((primary-fgs-self (primary-fgs self))
     (primary-obj (primary-exp obj)))
    (and
      primary-fgs-self
      (fgs-observation-p primary-obj)
      (not (same-ex-pointing
            primary-fgs-self
            primary-obj)))
    )
  )
  :description
  "Each alignment contains an FGS
   observation and the alignments have
   different pointings."
)

```

Figure 10:A Breaking Rule for Faint Guidance Sensor Observations

5.2. Example 2

This example is more complex, and demonstrates how higher level macros can be defined in LISP that use the merging shell facilities. Fig. 11 is an example of the use of such a macro, whose purpose is to prevent grouping of exposures that do not satisfy the “homogeneity criteria” into the same higher-level-object. The “homogeneity criteria” is defined to be the comparison of the values returned by a test that has, as its argument, an

exposure from the set of exposures being evaluated. If the values returned by the test for all exposures in the set are identical (a value of nil is considered to be identical with any other value), then the set is said to pass the homogeneity criteria, else it fails. The implementation of the macro itself involves implementation details that are beyond the scope of this paper.

```
(define-homogeneity-breaking-rule
  :name "BREAK PURE PARALLELS WITH INCOMPATIBLE TARGETS"
  :type :breaking
  :level '(al->ob)
  :priority 5
  :test (let ..<details intentionally left out>)
  :description
    "Pure parallel exposures (s.r. PARALLEL (but not PARALLEL WITH) or s.r. EXTERN PARALLEL WITH) should not be merged into the same obset with exposures with incompatible targets. Two exposures have targets incompatible for parallel merging if:
    1. Either is a solar system target.
    2. Either is an external target and the target names are different.
    3. One is pure parallel and the other is not."
)
```

Figure 11:An Example of the Use of the Homogeneity Breaking Rule

6. Output Products Generated

The merging shell enables the generation of two reports that have been found to be extremely useful by both the Users and Developers. The first is the "Merging Reasons Report". An excerpt from this report is shown in Fig. 12. This report enumerates the "reasons" why the observations were grouped in a certain way. The "reasons" are the names of the rules that were activated by the inferencing mechanism.

The second is due to the self-documenting feature of the shell, that creates a listing of all the rules that have been defined alongwith their explanations. An excerpt from such a listing is shown in Fig. 13. This listing is an integral part of the TRANS Scripting Guide, that contains exhaustive documentation on the requirements implemented within TRANS. The listing of the merging rules has been found to be very useful by the Configuration Management/Quality Assurance personnel charged with keeping the TSG up-to-date.

7. Validation of Manual Merging

As was pointed out earlier, an observation plan may be "manually merged" within TRANS by explicitly specifying the observation hierarchy that TRANS should use. This feature allows the user to circumvent the automatic merging mechanism within TRANS. Even while using "manual-merging" however, the user wants to be informed of all merging and breaking rules that may have been violated in selecting the specific manual-merge hierarchy. This task, which is called "validation of manual merging", is achieved within TRANS by first merging automatically using the merging and breaking rules, and then comparing the results to the specified manual-merging hierarchy. In case of conflict, the appropriate merging or breaking rule is output alongwith a diagnostic informing the user of a rule violation. An example of this diagnostic is shown in Fig. 14.

8. Implementation and Experience

As has already been mentioned, the TRANS Merging Shell was implemented in CLOS using an object-oriented paradigm. It should be pointed out, however, that TRANS is implemented in an extension of LISP called the transformation command language (XCL). XCL is implemented using the LISP macro facility, and supports a procedural rule syntax and allows abstraction for underlying data structures (Johnston and Gerb 1992). Hence, in order to

TRANSFORMATION VERSION DEVELOP 66.0

MERGING REASONS REPORT

GENERATED 12-7-1994 13:32:21

PROPOSAL 274 VERSION C

TRANSFORMED USING FULL-TRANS

SU 0027401:

OBSET 01:

BREAK REASON: NO MERGING RULE

ALIGNMENT 01:

BREAK REASON: NO MERGING RULE

EXPOSURE 01 (1.0000000):

BREAK REASON: NO MERGING RULE

ALIGNMENT 02:

MERGE REASON: MERGE ALIGNMENTS INTO OBSETS, PRIORITY 1

EXPOSURE 01 (2.0000000):

BREAK REASON: NO MERGING RULE

SU 0027402:

OBSET 02:

BREAK REASON: NO MERGING RULE

ALIGNMENT 01:

BREAK REASON: DONT CASUALLY MERGE DIFFERENT CONFIGS OR PRIORITIES, PRIORITY 2

EXPOSURE 01 (3.0000000):

BREAK REASON: NO MERGING RULE

<report truncated>

Figure 12: An Excerpt from a Merging Reasons Report

Merging Exposures into Alignments

This section of the Transformation Scripting Guide was created by the self documenting TRANS Merging Mechanism on 04/15/94, Version #21.

Rules for MERGING, Priority:8

Two exposures should be merged into the same alignment if:

1. MERGE PARALLEL WITH AND PRIMARY

Coordinated parallel exposures (exposures with the PARALLEL WITH sr) must be in the same alignment as their primary. This is not the case when coordinated parallels are being treated as pure (which does not apply to exposures with config=S/C).

2. MERGE PARALLEL WITH SAME PRIMARY

Exposures that are PARALLEL WITH the same primary should be merged into the same alignment. This is not the case when coordinated parallels are being treated as pure (which does not apply to exposures with config=S/C).

Rules for BREAKING, Priority: 5

Two EXPOSURES should be broken into different alignments if:

1. BREAK CONDS UNLESS DATA IS IDENTICAL

Exposures with the COND sr should never be merged with other exposures in the same SU unless they are conditional on exactly the same lines with exactly the same conditions (ignoring spaces).

2. BREAK COSTAR/AFM/POM UPLINKS FROM OTHER EXPOSURES

COSTAR exposures with the REQ UPLINK sr should be in their own obset. This is because mechanism history keeping cannot be performed with these exposures.

3. BREAK DIFFERENT WFPC/WFPC2 MODES

WFPC/WFPC2 exposures with different modes should be in different alignments.

4. BREAK EXPOSURES REQUIRING DIFFERENT POINTINGS

Exposures requiring different spacecraft pointings should be broken into separate alignments. Exposures are said to have the same pointing if the following two apply:

(a) They point to the same target. (Name must be the same. Same coordinates is not sufficient.)

(b) They have the same V2/V3 pointing. V2/V3 pointing is derived from the qaapertures table and the POS TARG offsets (see section 2.1 for details on computing V2/V3 pointing).

Figure 13: An Excerpt from the TRANS Scripting Guide Illustrating the Self Documenting Feature of the Merging Shell.

ERROR ENCOUNTERED

Exposure 15.000000 was merged into the previous scheduling unit due to manual merging.

This violates a BREAKING rule called

'BREAK ALIGNMENTS USING DIFFERENT GUIDE STARS' (priority 5).

DIAGNOSTIC TYPE: MERGING RULE VIOLATED DURING MANUAL MERGING

IN PROPOSAL 305

IN OBJECT EXPOSURE

Figure 14:An Example of a Manual Merging Validation Diagnostic

interface with the rest of TRANS, extensive use was made of XCL facilities for object creation and indexing. The examples in this paper all use CLOS syntax to refer to object slots and methods for ease of understanding. It was felt that introducing new syntax (XCL) in the examples was unnecessary.

A brief note on the necessity of having a separate mechanism for declaring slots for the data objects (separate from the actual class definitions of the data objects) is in order. The slot definition mechanism enables the following:

- use of arbitrarily complex symbolic expressions, the evaluations of which yield values for slot-value updates,
- specification of ordering in the updating of slot-values (important when there are inter-dependencies between slots),
- ease of adding new slots and modifying old ones, without having to deal with the implementation of the class definitions,
- separation of the declarative component (rules and slots) from the implementation.

The Merging Shell has been an integral part of the TRANS expert system for over 17 months. It has greatly simplified the modification of old rules and the declaration of new ones. The users are happy with its facilities for validation of manual-merging and report generation.

9. Conclusions

We have described here a declarative shell to facilitate organization of space observations into a hierarchy based on pre-specified rules for hierarchical organization. Even though the underlying principles of the shell are relatively simple, it offers a powerful way of expressing and dealing with knowledge related to the process of hierarchical organization. Its object-oriented implementation in CLOS provides for a seamless integration with a large expert system implemented in LISP. The concepts behind the merging shell can be applied to any space observation program where observations have to be organized into a hierarchical structure for purposes of planning and scheduling to satisfy resource constraints.

REFERENCES

- (Bose and Gerb 1994) Bose, A. and Gerb, A. Propagating Orientation Constraints for the Hubble Space Telescope, (to appear in) *J. of Robotics and Computer Integrated Manufacturing*.
- (Gerb 1991) Gerb, A. Transformation Reborn: A New Generation Expert System for Planning HST Operations, *J. of Telematics and Informatics*, v.8 n.4.
- (Johnston and Gerb 1992) Johnston, M.D. and Gerb, A. *Transformation Script Programmer Manual (Rev. K)*
- (Miller & Johnston 1991) Miller, G.E. and Johnston, M.D. A Case Study of Hubble Space Telescope Proposal Processing, Planning, and Long-Range Scheduling, *Proceedings AIAA-91*.
- (Steele 1990) Steele, G. *Common LISP, The Language* (2nd ed.), Bedford, MA: Digital Press.

1995720964

N95-27385

Stable Adaptive Neurocontrollers for Spacecraft and Space Robots

Robert M. Sanner*
University of Maryland at College Park
rmsanner@eng.umd.edu

5,0-63
47260
P-12

Abstract

This paper reviews recently developed techniques of adaptive nonlinear control using neural networks, and demonstrates their application to two important practical problems in orbital operations. An adaptive neurocontroller is first developed for spacecraft attitude control applications, and then the same design, slightly modified, is shown to be effective in the control of free-floating orbital manipulators. The algorithms discussed have guaranteed stability and convergence properties, and thus constitute viable alternatives to existing control methodologies. Simulation results are presented demonstrating the performance of each algorithm with representative dynamic models.

1 Introduction

Neural networks offer the potential for significantly extending the ability to control complex, poorly modeled dynamic systems. Unfortunately, however, connectionist control efforts often overlook the vast array of tools which have been developed in nonlinear systems theory, including adaptive techniques which are often much less complex than proposed neurocontrol solutions. Moreover, the crucial question of closed-loop stability is often ignored, or treated in an ad hoc fashion in connectionist control applications. Experienced control practitioners are thus often justifiably skeptical about the utility of proposed adaptive neurocontrollers.

*Dept. of Aerospace Eng. and Space Systems Lab.,College Park,MD 20742

Fortunately, however, it is possible to incorporate neural networks into the existing framework of nonlinear control and stability theory, and thereby develop designs which both advance the state of the art and possess guarantees of closed-loop stability and convergence. By uniting the new multivariable adaptive neurocontroller designs of (Sanner&Slotine,1992; Sanner&Slotine,1995) with recent work on adaptive and robust spacecraft attitude controllers (Bach&Paielli,1993; Dwyer&Siraramirez,1988; Egeland&Goddavn,1994; Slotine&Di Benedetto, 1990) we review below the construction of a stable neurocontroller for spacecraft attitude maneuvers. Noting then that the dynamics of a free-floating orbital robot have a structure mathematically similar to rigid spacecraft rotations (Papadopoulos,1990,1991), a similar adaptive neurocontrol methodology can be specified for these space robotic systems. Significantly, free-floating robotic systems cannot be treated in the context of "classic" nonlinear adaptive systems theory, and thus adaptive neurocontrollers represent an important new enabling technology in space robotics.

Section 2 first discusses available nonlinear control techniques for spacecraft attitude maneuvers, then demonstrates how adaptive neural networks can be used to significantly extend these methods when faced with relatively unstructured uncertainty about nature of the torques influencing the motion of the spacecraft. In Section 3, the same neurocontrol design, slightly modified, is shown to be effective in the control of free-floating orbital manipulators. Each section provides a complete specification of the structure of the control and adaptation laws, and provides simulation results which demonstrate the per-

formance of the controller on representative systems.

2 Attitude control

2.1 Problem Statement

The attitude dynamics of a rigid spacecraft subject to torques applied by gas jet thrusters can be written as (Hughes, 1986)

$$\mathbf{H}\dot{\boldsymbol{\omega}} - \mathbf{S}(\mathbf{H}\boldsymbol{\omega})\boldsymbol{\omega} = \boldsymbol{\tau} \quad (1)$$

$$\dot{\mathbf{C}} = -\mathbf{S}(\boldsymbol{\omega})\mathbf{C} \quad (2)$$

where \mathbf{H} is the constant, symmetric, positive definite spacecraft inertia matrix, \mathbf{C} is the rotation matrix which describes the attitude of the vehicle with respect to an inertial frame, and $\boldsymbol{\omega}$ is the angular velocity of the spacecraft with respect to this frame. The vector $\boldsymbol{\tau}$ represents the torques applied to the spacecraft by its attached attitude control thrusters. In these equations, \mathbf{S} provides the matrix representation of the cross product operator, so that $\mathbf{a} \times \mathbf{b} = \mathbf{S}(\mathbf{a})\mathbf{b}$, and hence

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Given measurements of the current vehicle attitude and angular velocity, the goal of the attitude control problem is to design a feedback control law for the torques, $\boldsymbol{\tau}$, which will ensure that the actual attitude will asymptotically track a desired attitude, defined by

$$\dot{\mathbf{C}}_d = -\mathbf{S}(\boldsymbol{\omega}_d)\mathbf{C}_d, \quad (3)$$

where $\boldsymbol{\omega}_d$ is a specified desired angular velocity, assumed to be bounded, with a bounded derivative.

While the elements of the direction cosine matrix can be used directly to develop suitable control laws (Bach&Paielli, 1993), more compact and computationally efficient algorithms can be developed by instead utilizing the quaternion representation of vehicle attitude. In this formulation, vehicle attitude is specified by a three element vector, \mathbf{e} , and a scalar parameter, η , collected together into the four element quaternion, \mathbf{e} , defined so that

$$\mathbf{e} = \begin{bmatrix} \mathbf{e} \\ \eta \end{bmatrix} = \begin{bmatrix} \sin \frac{\varphi}{2} \mathbf{a} \\ \cos \frac{\varphi}{2} \end{bmatrix}.$$

Here \mathbf{a} is the unit eigenaxis of the rotation from the inertial to the body frame, i.e. $\mathbf{a} = \mathbf{C}\mathbf{a}$, and φ is the magnitude of the rotation about this axis (Hughes, 1986). More explicitly, the elements of the quaternion completely determine the rotation matrix \mathbf{C} through the relation $\mathbf{C} = \mathbf{R}(\mathbf{e})$, where

$$\mathbf{R}(\mathbf{e}) = (\eta^2 - \mathbf{e}^T \mathbf{e})\mathbf{I} + 2\mathbf{e}\mathbf{e}^T - 2\eta\mathbf{S}(\mathbf{e}).$$

Finally, in place of (2), the kinematics of the quaternion representing the vehicle attitude is given by

$$\dot{\mathbf{e}} = \mathbf{J}(\mathbf{e})\boldsymbol{\omega} \quad (4)$$

where

$$\mathbf{J}(\mathbf{e}) = \frac{1}{2} \begin{bmatrix} \eta\mathbf{I} + \mathbf{S}(\mathbf{e}) \\ -\mathbf{e}^T \end{bmatrix}.$$

A similar equation defines the evolution of the desired attitude, $\dot{\mathbf{e}}_d = \mathbf{J}(\mathbf{e}_d)\boldsymbol{\omega}_d$.

In order to develop a feedback control strategy for this system, an appropriate measure of attitude error must be synthesized. Using the actual and desired rotation matrices, a natural measure for this purpose can be defined as

$$\tilde{\mathbf{C}} = \mathbf{C}\mathbf{C}_d^T. \quad (5)$$

With this definition, $\tilde{\mathbf{C}}$ is the matrix which transforms a vector in the desired frame to one in the body frame, and in particular, when $\mathbf{C} = \mathbf{C}_d$, $\tilde{\mathbf{C}} = \mathbf{I}$. The dynamics of this error measure are easily computed from the actual and desired attitude dynamics

$$\begin{aligned} \dot{\tilde{\mathbf{C}}} &= \dot{\mathbf{C}}\mathbf{C}_d^T + \mathbf{C}\dot{\mathbf{C}}_d^T \\ &= -\mathbf{S}(\boldsymbol{\omega})\tilde{\mathbf{C}} + \tilde{\mathbf{C}}\mathbf{S}(\boldsymbol{\omega}_d) \\ &= -\mathbf{S}(\tilde{\boldsymbol{\omega}})\tilde{\mathbf{C}} \end{aligned} \quad (6)$$

where $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} - \tilde{\mathbf{C}}\boldsymbol{\omega}_d$.

Alternatively, using the quaternion representation one obtains $\tilde{\mathbf{C}} = \mathbf{R}(\tilde{\mathbf{e}}) = \mathbf{R}(\mathbf{e}\mathbf{e}_d^{-1})$, where the inverse of a quaternion is defined as

$$\mathbf{e}^{-1} = \begin{bmatrix} -\mathbf{e} \\ \eta \end{bmatrix}$$

and quaternion multiplication is defined so that $\mathbf{e}_2\mathbf{e}_1 = \mathbf{U}(\mathbf{e}_1)\mathbf{e}_2$ with

$$\mathbf{U}(\mathbf{e}) = \begin{bmatrix} \eta\mathbf{I} + \mathbf{S}(\mathbf{e}) & \mathbf{e} \\ -\mathbf{e}^T & \eta \end{bmatrix}.$$

Using (4) and (6), the quaternion error dynamics can be computed as

$$\dot{\tilde{\mathbf{e}}} = \mathbf{J}(\tilde{\mathbf{e}})\tilde{\boldsymbol{\omega}}. \quad (7)$$

Note that the quaternion measure of attitude error, $\tilde{\mathbf{e}}$, admits the same interpretation as $\tilde{\mathbf{C}}$. In particular, $\tilde{\mathbf{e}}$ is the quaternion corresponding to the attitude of the actual frame with respect to the desired vehicle frame, and when the two frames are aligned $\tilde{\mathbf{e}}^T = [0, 0, 0, \pm 1]$.

2.2 Conventional fixed and adaptive controller designs

Several authors have exploited the structure of the above equations to develop effective nonlinear control strategies which solve the tracking problem posed (Bach&Paielli,1993; Egeland&Godhavn,1994; Fossen,1992; Paielli&Bach,1993; Wen&Kreutz-Delgado,1991; Wie&Barba,1985; Wie et al.,1989). Most recently, (Egeland&Godhavn,1994), building upon the fundamental results of (Slotine&Li,1987; Slotine&Di Benedetto,1990), have proposed a particularly compact algorithm which is especially amenable to adaptive operation. The current section reviews this new algorithm, in preparation for the neurocontrol extensions considered in the following section.

The algorithm of (Egeland&Godhavn,1994) utilizes the composite error metric

$$\mathbf{s} = \tilde{\boldsymbol{\omega}} + \lambda\tilde{\mathbf{e}} = \boldsymbol{\omega} - \boldsymbol{\omega}_r \quad (8)$$

where

$$\boldsymbol{\omega}_r = \tilde{\mathbf{C}}\boldsymbol{\omega}_d - \lambda\tilde{\mathbf{e}} \quad (9)$$

and $\lambda > 0$ is an arbitrary positive constant. Provided that the system inertia matrix, \mathbf{H} , is known precisely, the control law

$$\boldsymbol{\tau}(t) = -\mathbf{K}_D(t)\mathbf{s}(t) + \boldsymbol{\tau}^{nl}(t), \quad (10)$$

where $\mathbf{K}_D(t)$ is a uniformly positive definite matrix and

$$\boldsymbol{\tau}^{nl} = \mathbf{H}\dot{\boldsymbol{\omega}}_r - \mathbf{S}(\mathbf{H}\boldsymbol{\omega})\boldsymbol{\omega}_r,$$

can then be shown to produce asymptotically convergent closed-loop tracking of any desired attitude trajectory, given by \mathbf{e}_d and $\boldsymbol{\omega}_d$.

Under the more realistic assumption that there is some initial uncertainty about the actual distribution of mass in the spacecraft, the above algorithm

can be modified to continuously tune the nonlinear component $\boldsymbol{\tau}^{nl}$, thus adaptively compensating for this uncertainty. Implementation of this modification requires first factoring the nonlinear components of the control law:

$$\begin{aligned} \boldsymbol{\tau}^{nl} &= \mathbf{H}\dot{\boldsymbol{\omega}}_r - \mathbf{S}(\mathbf{H}\boldsymbol{\omega})\boldsymbol{\omega}_r \\ &= \mathbf{Y}(\boldsymbol{\omega}, \boldsymbol{\omega}_r, \dot{\boldsymbol{\omega}}_r)\mathbf{a} \end{aligned} \quad (11)$$

where \mathbf{a} contains the 6 unique elements of the spacecraft inertia matrix. Using this factorization, but perhaps lacking exact knowledge of the mass properties of the spacecraft, the nonlinear components can be implemented using *estimates*, $\hat{\mathbf{a}}$, of the true mass properties, \mathbf{a}

$$\boldsymbol{\tau} = -\mathbf{K}_D\mathbf{s} + \mathbf{Y}\hat{\mathbf{a}}. \quad (12)$$

By then continuously tuning these estimates according to the adaptation law

$$\dot{\hat{\mathbf{a}}} = -\boldsymbol{\Gamma}\mathbf{Y}^T\mathbf{s}, \quad (13)$$

where $\boldsymbol{\Gamma}$ is a constant, symmetric positive definite matrix controlling the rate of adaptation, (Egeland&Godhavn,1994) show that the resulting closed-loop system is stable, and again guarantees asymptotically perfect tracking of any smooth desired attitude trajectory.

Substantial prior knowledge about the rotational dynamics must be utilized in order to separate the nonlinear functions comprising the elements of \mathbf{Y} , from the mass parameters \mathbf{a} ; such a parameterization is readily obtained for the idealized rigid body dynamics of a spacecraft. More complete models of the rotational dynamics, however, may also include a variety of environmental torques, arising from gravity gradients, solar pressure, magnetic fields, and atmospheric drag, to name the more significant sources, which may not readily admit such a convenient parameterization of uncertainty. Indeed, in many cases the actual physics underlying the structure of the environmental torques may be too complex or too poorly understood to provide an explicit, closed-form description of their impact on the rotational dynamics. Moreover, by “hardcoding” into \mathbf{Y} a description of the expected environment, through the choice of specific functions assumed to model these torques, the system becomes excessively “rigid”, incapable of responding appropriately to unexpectedly different environments.

In order to address this more general uncertainty model, the next section reviews techniques whereby the established functional approximation abilities of “neural” networks (Cybenko,1989; Girosi&Poggio,1990; Hornik et al.,1989) can be employed to provide the flexibility necessary to compensate for uncertainty on the actual component *functions* in appearing in the dynamic model.

2.3 Adaptive control and neural nets

Incorporating the above sources of environmental torques, a more complete model of the rotational dynamics might thus be

$$\mathbf{H}\dot{\boldsymbol{\omega}} - \mathbf{S}(\mathbf{H}\boldsymbol{\omega})\boldsymbol{\omega} + \mathbf{E}(\mathbf{e}, \boldsymbol{\omega}) = \boldsymbol{\tau}, \quad (14)$$

where the vector \mathbf{E} now contains any torques applied to the vehicle by its environment. If the structure of these new torques were known explicitly, by augmenting $\boldsymbol{\tau}^{nl}$ in (??) with the term $\mathbf{E}(\mathbf{e}, \boldsymbol{\omega})$, the resulting closed-loop system would again provide asymptotically convergent tracking. However, unlike the situation addressed by the algorithms of the previous section, where there was uncertainty only about the mass properties of the spacecraft, in this section the functional form of the torques appearing in the spacecraft dynamic model, both the rigid body and the environmental torques, is assumed to be completely unknown. The required $\boldsymbol{\tau}^{nl}$ can hence not be implemented, nor can the above adaptive technique be used to learn the required $\boldsymbol{\tau}^{nl}$, since by assumption the prerequisite $\mathbf{Y}\mathbf{a}$ parameterization is unknown or impossible to obtain.

Proceeding similarly to (Sanner&Slotine,1995), consider instead the following alternative representation of the nonlinear component of the required control input:

$$\begin{aligned} \boldsymbol{\tau}^{nl} &= \mathbf{H}\dot{\boldsymbol{\omega}}_r - \mathbf{S}(\mathbf{H}\boldsymbol{\omega})\boldsymbol{\omega}_r + \mathbf{E}(\mathbf{e}, \boldsymbol{\omega}) \\ &= \mathbf{M}(\mathbf{x})\mathbf{v} \end{aligned} \quad (15)$$

or, in component form,

$$\tau_i^{nl} = \sum_{j=1}^7 M_{i,j}(\mathbf{x}) v_j$$

where $\mathbf{v}^T = [\dot{\boldsymbol{\omega}}_r^T, \boldsymbol{\omega}_r^T, 1]$, and for notational convenience, the components of the vehicle state have been

collected into a single vector $\mathbf{x}^T = [\mathbf{e}^T, \boldsymbol{\omega}^T]$. Unlike expansion (11), which decomposes $\boldsymbol{\tau}^{nl}$ into a matrix of known functions, \mathbf{Y} , multiplying a vector of unknown constants \mathbf{a} , this expansion decomposes $\boldsymbol{\tau}^{nl}$ into a 3×7 matrix of unknown functions \mathbf{M} , multiplying a vector of 7 known signals \mathbf{v} .

Without the ability to determine a $\mathbf{Y}\mathbf{a}$ factorization, an adaptive controller capable of producing the required control input must instead learn each of the 21 unknown component *functions*, $M_{i,j}(\mathbf{x})$, as opposed to the conventional model which must learn only unknown *constants*, a . In the robotic applications considered in (Sanner&Slotine,1995), the controller implements estimates of these functions using adaptive neural networks. Indeed, since the components of $\mathbf{S}(\boldsymbol{\omega})\mathbf{H}$ are continuous functions of their arguments, if the same also is true of the environmental forces, \mathbf{E} , such networks can be used to uniformly approximate to a chosen accuracy each component function of \mathbf{M} on any closed, bounded subset, A , of the state space (Cybenko,1989; Girosi&Poggio,1990; Hornik et al., 1989).

Thus, if the functions in \mathbf{M} are sufficiently smooth, a neural network approximation of the form

$$\tau_i^N(t) = \sum_{j=1}^7 N_{i,j}(\mathbf{x}(t), \mathbf{p}) v_j(t) \quad (16)$$

can accurately approximate the required nonlinear control input for appropriate values of the free network parameters \mathbf{p} . Here each $N_{i,j}$ is an output of a single hidden layer neural network of the form

$$N_{i,j}(\mathbf{x}, \mathbf{p}) = \sum_{k=1}^N c_{i,j,k} g_k(\mathbf{x}, \xi_k),$$

and the neural approximation theorems ensure that, for several different neural computation models, g_k , there exist values of the free parameters N , $c_{i,j,k}$ and ξ_k , which will approximate the continuous functions in \mathbf{M} to a chosen level of uniform accuracy on a compact set A . In this control setting, defining $d = \boldsymbol{\tau}^{nl} - \boldsymbol{\tau}^N$, one thus has that for proper choice of N , $c_{i,j,k}$ and ξ_k

$$|d_i(\mathbf{x}, \mathbf{v})| \leq \sum_{j=1}^7 \delta_{i,j} |v_j|$$

for any point $\mathbf{x} \in A$, where each $\delta_{i,j}$ is the worst case error of the network approximation to $M_{i,j}$ on the

set A . Provided that $\omega_d(t)$ and $\dot{\omega}_d(t)$ are bounded, as above, over this subset of the state space, the discrepancy between the “neural” approximation and the required nonlinear terms can thus be made arbitrarily small by appropriate design of the network employed (Sanner,1993; Sanner&Slotine,1995).

The network used in (16) has the 7 components of the vehicle attitude state as its inputs, and 21 outputs representing the approximations to each $M_{i,j}(\mathbf{x})$. While in principle, each of the independent network parameters, N , ξ_k , and $c_{i,j,k}$ could be learned, new theoretical results on constructive neural approximation techniques provide a variety of algorithms for effectively preselecting certain of the network parameters based upon estimates of the *smoothness* of the functions being approximated. For example, for certain radial basis function network models (Broomhead&Lowe,1988; Poggio&Girosi,1990), i.e. networks for which $g_k(\mathbf{x}, \xi_k) = g(\sigma_k \|\mathbf{x} - \xi_k\|)$ for a positive scaling parameter σ_k , the parameters ξ_k can be chosen to encode a uniform mesh over the set A whose spacing is determined by bounds on the significant frequency content of the Fourier transform of the functions being approximated (Sanner&Slotine,1992).

This analysis, and similar constructive techniques, leaves only the specific output weights, $c_{i,j,k}$, to be learned in order to accurately approximate the particular functions of the assumed smoothness class which appear in the matrix M . The following section reviews how the techniques of (Sanner&Slotine,1995,Sanner&Slotine,1992) can be applied to the attitude control problem, specifying a neurocontrol law and adaptation mechanism which can stably learn these required output weights, producing asymptotically convergent tracking of a desired attitude. For a more detailed analysis of this algorithm, including more general adaptation methods, refer to (Sanner&Slotine,1995).

2.4 Adaptive neurocontroller designs

Despite their potential, practical implementations of neural computation models are at best capable of providing only *locally approximate* representations of the required control input. Use of such a device in place of explicit, prior knowledge about the dynamic structure thus introduces the unmeasur-

able disturbance, \mathbf{d} , into the closed-loop dynamic model. Since \mathbf{d} is generally nonvanishing, the adaptive system must be robust to this perturbation, lest it cause the closed-loop system to become unstable (Reed&Ioannou,1989;Sanner&Slotine,1992).

To accommodate the required robust modifications, first define a set $A_d \subset \mathcal{R}^7$ containing the trajectories the system must follow, a closed and bounded “nominal operating range” $A \supset A_d$, and a smooth modulation function, $m(t)$, which is unity outside the set A , vanishes inside A_d and otherwise satisfies $0 < m(t) < 1$. Notice that A_d can be chosen as the cartesian product of the four dimensional cube $[-1, 1]^4$ and a three dimensional cube containing $\omega_d(t)$ for all t , since by definition the quaternion components only assume values in $[-1, 1]$.

The proposed adaptive control law can then be written as

$$\tau(t) = -\mathbf{K}_D(t) \mathbf{s}(t) + m(t) \tau^{sl}(t) + (1 - m(t)) \hat{\tau}^N(t) \quad (17)$$

where the robust *sliding controller* component is $\tau^{sl}(t) = -K_i(\mathbf{x}, t) \operatorname{sgn}(s_i(t))$, whose gains are chosen, similar to the designs in (Slotine&Li,1991; Dwyer&Sira-Ramirez,1988), so that

$$K_i(\mathbf{x}, t) \geq \sum_{j=1}^7 |M_{i,j}(\mathbf{x}) v_j(t)|.$$

These upper bounds, which can be quite loose, are assumed to be available *a priori*.

Assuming a network architecture has been selected on the basis of the assumed smoothness of the functions required in the control law, the adaptive neural component of the controller is given by

$$\hat{\tau}_i^N(t) = \sum_{j=1}^7 \sum_{k=1}^N \hat{c}_{i,j,k}(t) g_k(\mathbf{x}(t), \xi_k) v_j(t). \quad (18)$$

Building from the results in (Sanner,1993; Sanner&Slotine,1995), (Sanner&Vance,1994) show that the control law (17), (18) coupled with the continuous network learning rule

$$\dot{\hat{c}}_{i,j,k}(t) = \mathcal{P}(-\gamma_{i,j,k} s_i(t) v_j(t) g_k(\mathbf{x}(t), \xi_k), \hat{c}_{i,j,k}(t), \bar{c}_{i,j,k}) \quad (19)$$

will produce a stable closed-loop system and asymptotic tracking of any desired attitude with an ultimate accuracy limited only by the network approximation capabilities, $\delta_{i,j}$. Here $\bar{c}_{i,j,k}$ is an upper

bound on the magnitude of each required output weight, and the projection operator \mathcal{P} is defined so that $\mathcal{P}(x, y, z) = (1-m)x$ if $-z < y < z$, or if $y \leq -z$ and $x > 0$, or if $y \geq z$ and $x < 0$; $\mathcal{P}(x, y, z) = 0$ otherwise.

This robust adaptation mechanism effectively restricts the search for the required weights to a subset of the $21N$ dimensional weight space, preventing the possibly unbounded "wandering" which can be provoked by the disturbance d (Slotine&Li,1991; Reed&Ioannou,1989). The robust controller component, τ^{st} , is a supervisory mechanism which, if required, will stabilize the system in its initial learning phases, smoothly returning the state to its nominal operating range, on which the network is capable of well approximating M .

2.5 Attitude control example

This section demonstrates the performance of the proposed algorithm on a simulated attitude control problem. The spacecraft inertia matrix used in the simulation is

$$H = \begin{bmatrix} 60 & 5 & 0 \\ 5 & 78 & 10 \\ 0 & 10 & 38 \end{bmatrix},$$

and the desired attitude trajectory used to evaluate the controller was specified by

$$\begin{aligned} \omega_{d,1} &= \frac{-3(\cos t + 3\sqrt{3} \sin t)}{8\sqrt{2}r(t)} \\ \omega_{d,2} &= \frac{3(5 \cos t - \sqrt{3} \sin t)}{8\sqrt{2}r(t)} \\ \omega_{d,3} &= \frac{3(\sqrt{3} \cos t + \sin t)}{8r(t)} \\ \dot{\mathbf{e}}_d &= J(\mathbf{e}_d)\omega_d \end{aligned}$$

where $r(t) = 1 + .2 \cos t$. To implement the control law (17), the tracking error metric s is computed using (21) with $\lambda = 10$, and the gains $K_D = 100I$ are used for the linear feedback components.

Given the definition of the desired trajectory, the nominal operating range, A_d , was chosen as $A_d = [-1, 1]^4 \times [-1.75, 1.75]^3$. The neural network, \mathcal{N} , employed in the control law uses radial gaussian nodes, with $g_k(\mathbf{x}, \xi_k) = \exp(-\sigma_k \|\mathbf{x} - \xi_k\|^2)$ to approximate the functions in M on the set $A = [-1.1, 1.1]^4 \times$

$[-2, 2]^3$. For simplicity in this simulation, the network was designed assuming that any applied environmental forces are a function of ω only. Under these conditions, M is also a function of ω only, and the resulting network requires only the three inputs, ω_i , and still 21 outputs, $\mathcal{N}_{i,j}$. Using the constructive analysis techniques in (Sanner,1993) to initially fix some of the network structure, each node uses the same scaling parameter, $\sigma_k = 6$, and the gaussian "centers" ξ_k lie on a regular lattice of mesh size $\Delta = 0.5$ covering the set $[-2.5, 2.5]^3$. There are thus a total of 1,331 gaussian nodes and 27,951 output weights which the network must learn in order to accurately approximate the elements of M .

Each output weight was initialized to zero, simulating an initial total lack of knowledge about the dynamics of the system. During the simulation, these weights were continuously updated according to the learning rule (19) together with the adaptation gains $\gamma_{i,j,k} = 2.5$ for each i, j, k . The upper bounds $\bar{c}_{i,j,k} = 200$ were used to implement the projection mechanism. The modulation function, $m(t)$, and sliding controller gains were chosen as in (Sanner&Slotine,1992; Sanner&Slotine,1995). In this particular example, however, the supervisory action of the sliding controller was never needed.

Figure 1 shows the performance of the algorithm using this network, when the spacecraft attitude evolves according to the ideal model (1). After a transient period the attitude tracking errors, $\tilde{\epsilon}_i$, are reduced to a small neighborhood of zero, and $\tilde{\eta}$ converges to near 1, indicating that the spacecraft is asymptotically tracking the desired attitude. For comparison, Figure 2 illustrates the tracking which would be obtained without use of the adaptive network, thus implementing a quaternion "PD" type control strategy. The initial performance of the network is virtually identical to the "PD" algorithm, but the network performance rapidly becomes markedly superior.

Figure 3 shows the performance of the algorithm, using the same network and initialization, when the spacecraft attitude instead evolves according to (14), where the environmental torques are given by

square-law drag terms of the form

$$\mathbf{E}(\omega) = \begin{bmatrix} 8|\omega_1| & 0 & 0 \\ 0 & 15|\omega_2| & 0 \\ 0 & 0 & 25|\omega_3| \end{bmatrix} \omega.$$

Note that while this particular environmental torque is not common in an orbital environment, it is constitutes a significant influence in neutral buoyancy simulation of orbital operations. The large perturbations these representative hydrodynamic torques introduce to the ideal rigid body dynamics provide a significant additional dynamic component which must be learned by the neural network. As Figure 3 shows, however, the ultimate tracking performance obtained in the presence of these torques is virtually identical to that obtained with the unperturbed dynamics, indicating that the network is successfully compensating for the new dynamic components. By comparison, if the adaptive contribution of the network is omitted in the control law, the tracking performance is significantly degraded, as demonstrated in Figure 4.

3 Free-floating robot control

3.1 Problem statement and neurocontrol solutions

When a robotic arm is mounted to the front of a submersible or orbital vehicle, the motion of the arm will couple to that of its mobile base. If the base is allowed to rotate as the arm moves, that is, if no torques are directly applied to the base allowing it to resist the induced motion, the resulting robotic system is termed a *free-floating* manipulator. Such systems are especially attractive in space operations, where worksite damage could ensue from use of a propulsion system, and where avoiding the use of reaction mass may make the mission potentially more affordable by reducing launch costs and/or extending the useful life of the system.

A careful analysis of the coupled dynamics of a manipulator arm mounted on a free-floating base shows that the spacecraft attitude states may be eliminated from the coupled equations, resulting in a compact set of differential equations describing the motion of arm joints. These equations have the same the same general form as the equations of motion for

fixed-base manipulators (Papadopoulos,1990,1991), i.e.

$$\mathbf{H}^*(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{F}^*(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{E}^*(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau}_m. \quad (20)$$

In this equation, $\mathbf{q} \in \mathcal{R}^n$ is an n vector of manipulator joint angles, \mathbf{H}^* is a symmetric, uniformly positive definite inertia matrix, and \mathbf{F} is a matrix accounting for the centripetal and Coriolis forces arising from the arm motions. The vector $\boldsymbol{\tau}_m$ represents the torque applied by motors at each manipulator joint. Finally, \mathbf{E}^* again represents the effect of any additional environmental forces.

In addition to the similarities to fixed-base manipulator dynamics, (20) is clearly also quite similar to the spacecraft rotation models examined above. Indeed, formally combining the spacecraft kinematic and dynamic equations produces a differential equation structurally identical to (20) (Slotine&Di Benedetto,1990). It is precisely this structural equivalence which has inspired the recent adaptive attitude control algorithms (Egeland&Goddavn,1994; Fossen,1992; Slotine&Di Benedetto,1990), including the one reviewed above, from the fundamental robotic result presented in (Slotine&Li,1987).

This suggests moreover that the adaptive neurocontroller presented above can also be used to cause the joint angles of a free-floating manipulator to asymptotically track any desired sequence of joint angles, \mathbf{q}_d . By redefining the tracking error metric

$$s = \dot{\tilde{\mathbf{q}}} + \lambda \tilde{\mathbf{q}} \quad (21)$$

where now $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$, (Sanner&Vance,1994) show that the preceding adaptive neurocontrol algorithm indeed provides a stable closed-loop system and asymptotic convergence of the tracking errors to a small neighborhood of zero. In such applications, the network inputs are the states of the robotic arm, $\mathbf{x}^T = [\mathbf{q}^T, \dot{\mathbf{q}}^T]$, and the auxiliary signals are $\mathbf{v}^T = [\tilde{\mathbf{q}}_r^T, \dot{\tilde{\mathbf{q}}}_r^T, 1]$, where $\dot{\tilde{\mathbf{q}}}_r = \dot{\mathbf{q}}_d - \lambda \tilde{\mathbf{q}}$.

An additional design simplification can be obtained in these robotic applications by noting that the centripetal and Coriolis forces are quadratic in velocity. If also \mathbf{E}^* is a function of \mathbf{q} only, or can be decomposed as $\mathbf{E}^*(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{E}_0^*(\mathbf{q})\mathbf{f}(\dot{\mathbf{q}})$, where $\mathbf{f}(\dot{\mathbf{q}})$ represents a known $\dot{\mathbf{q}}$ dependence, the neural component of the controller can be chosen as

$$\hat{\tau}_i^N(t) = \sum_j \sum_{k=1}^N \hat{c}_{i,j,k}(t) g_k(\mathbf{q}(t), \xi_k) w_j(t). \quad (22)$$

The new auxiliary signals, w_j , are respectively the components of $\ddot{\mathbf{q}}_r$, $[\dot{\mathbf{q}}\dot{\mathbf{q}}_r]$, and $\mathbf{f}(\dot{\mathbf{q}})$ (or simply 1 for the latter component if \mathbf{E}^* is a function of \mathbf{q} only). The notation $[\dot{\mathbf{q}}\dot{\mathbf{q}}_r]$ is a shorthand for all possible combinations $\dot{q}_i \dot{q}_{r,j}$ for each $i, j = 1, \dots, n$. With this use of the network in the controller, the adaptation mechanism is modified to be

$$\dot{\hat{c}}_{i,j,k}(t) = \mathcal{P}(-\gamma_{i,j,k}s_i(t) w_j(t) g_k(q(t), \xi_k), \hat{c}_{i,j,k}(t), \bar{c}_{i,j,k}). \quad (23)$$

Despite the gross structural similarity of the dynamics (20) to both fixed-base manipulator dynamics and the spacecraft rotation models (14),(4), there are important differences in the nature of the functions which appear. In particular, the matrices \mathbf{H}^* and \mathbf{F}^* in the free-floating manipulator dynamics are significantly more complex than their counterparts in spacecraft or fixed-base robot dynamics. These matrices are so complex, in fact, that in the face of uncertainty about the manipulator mass properties, the parameterization $\tau^{nl} = \mathbf{Y}\hat{\mathbf{a}}$ is simply *not possible* even for the ideal ($\mathbf{E}^* = \mathbf{0}$) dynamics of free-floating manipulator systems (Papadopoulos,1990; Sanner&Vance,1994). This is in marked contrast to spacecraft and fixed-base robot dynamics, and provides a specific example of a situation in which the adaptive function approximations implemented by neural networks yield a new solution to an otherwise intractable control problem.

3.2 Free-floating robot example

Figure 5 shows a 2 link manipulator attached to a spacecraft, with both spacecraft and arm motion restricted to a single plane. The 3 independent degrees of freedom of the system are θ , the orientation of the spacecraft with respect to an inertial reference frame, q_1 and q_2 which respectively describe the relative orientation of the first manipulator link to the spacecraft and the second link to the first link.

For simplicity, the simulation assumes an ideal dynamic model with $\mathbf{E}^* = \mathbf{0}$ in (20). Figure 6 gives the mass, inertia, and relevant dimensions for the system. The centers of mass of the spacecraft and of each link are located centrally, as indicated in Figure 5. To demonstrate the performance of the proposed neurocontroller, the desired trajectory was $q_{d,1}(t) = 1.2 \cos(0.8t)$ and $q_{d,2}(t) = 0.5 \cos(2.1t)$. Given the definition of the trajectories the system

is required to follow, the set A_d was chosen as $A_d = [-1.2, 1.2] \times [-0.5, 0.5] \times [-1, 1] \times [1.05, 1.05]$, and the nominal operating range, A was chosen as $A = [-1.4, 1.4] \times [-0.6, 0.6] \times [-1.1, 1.1] \times [1.2, 1.2]$.

Using the simplified controller with (22) above, the neural network employed in the control law has the 2 inputs $q_1(t)$ and $q_2(t)$, and 12 outputs. The network used for the simulation again employs radial gaussian nodes in the hidden layer, with each gaussian center arranged on a regular lattice of mesh size $\Delta = 0.2$ covering the set $[-2, 2] \times [-1.4, 1.4]$. Each node again used the same scale factor, here taken as $\sigma_k = 13$. There are thus a total of 315 gaussian nodes and 3780 output weights which the network must learn in order to accurately approximate the required control input.

Each output weight was again initialized to zero, and continuously updated according to the learning rule (23) together with the adaptation gains $\gamma_{i,j,k} = 2$ for each i, j, k . The error metric s is computed using (21) with $\lambda = 10$, and the gains $\mathbf{K}_D = 10\mathbf{I}$ are used for the linear feedback control components. Finally, the modulation function and sliding gains were again computed as in (Sanner&Slotine,1995; Sanner&Vance,1994).

Figure 7 displays the performance of the neurocontroller tracking the specified joint space trajectory. After a brief initial transient, the tracking errors in each joint converge to a small neighborhood of zero. Compare this with the performance of the “PD” controller obtained by omitting the contribution of the adaptive network from the control law. Although initially (before any learning has occurred) the performance of the neurocontroller resembles that of the pure “PD” controller, the neurocontroller gradually reduces the tracking error, eventually achieving worst case error a factor of 20 smaller than those obtained with the PD controller.

4 Concluding remarks

High performance control of orbital robots and spacecraft is an essential technology to ensure that these systems will be truly useful in future orbital operations. Most importantly, the accuracy and reliability of the algorithms employed must be assured, even in the face of real-world uncertainty on the physical properties of the system. In this paper we

have demonstrated that, far from academic curiosities, adaptive "neural" networks provide unique solutions for important practical problems in the control of spacecraft and space robots, which otherwise are difficult to solve with established adaptive control techniques. The stability and convergence properties of the algorithms described provide the assurances of reliability and effectiveness needed to make such controllers viable alternatives to existing control algorithms.

References

- [1] Bach, R., and Paielli, R., "Linearization of attitude-control error dynamics", *IEEE Trans. Aut. Control*, 38(10), 1521-1525.
- [2] Broomhead, D. S., and Lowe, D., "Multivariable functional interpolation and adaptive networks", *Complex Systems*, 2, 321-355, 1988.
- [3] Cybenko, G., "Approximations by superposition of a sigmoidal function", *Math. Cont., Sig, and Sys.*, vol. 2, 303-314, 1989.
- [4] Dwyer, T., and Sira-Ramirez, H., "Variable-structure control of spacecraft attitude maneuvers", *J. Guid., Contr., and Dyn.*, 11(3), pp. 262-270, 1988.
- [5] Egeland, O., and Godhavn, J.-M., "Passivity-based adaptive attitude control of a rigid spacecraft", *IEEE Trans. Aut. Control*, 39(4), pp. 842-845, 1994.
- [6] Fossen, T. I., *Guidance and Control of Ocean Vehicles*, Wiley and Sons, New York, NY, 1994.
- [7] Funahashi, K., "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, 2, 183-192, 1989.
- [8] Girosi, F., and Poggio, T., "Networks and the best approximation property", *Biological Cybernetics*, 63, 169-176, 1990.
- [9] Hornik, K., Stinchcombe, M., and White, H., "Multilayer feedforward networks are universal approximators", *Neural Networks*, 2, 359-366, 1989.
- [10] Hughes, P. C., *Spacecraft Attitude Dynamics*, Wiley and Sons, New York, NY, 1986.
- [11] Paielli, R., and Bach, R., "Attitude control with realization of linear error dynamics", *J. Guid. Control and Dyn.*, 16(1), 182-189, 1993.
- [12] Papadopoulos, E. G., *On the Dynamics and Control of Space Manipulators*, Ph.D. Thesis, MIT Dept. of Mech. Eng., Cambridge, MA, 1990.
- [13] Papadopoulos, E. G., "On the nature of control algorithms for free-floating space manipulators", *IEEE Trans. Rob. Aut.*, 7(6), 750-758, 1991.
- [14] Poggio, T., and Girosi, F., "Networks for approximation and learning", *Proc. IEEE*, 78(9), 1481-1497, 1990.
- [15] Reed, J. S., and Ioannou, P. A., "Instability analysis and robust adaptive control of robotic manipulators", *IEEE Trans. Rob. and Aut.*, 5, 381-386, June 1989.
- [16] Sanner, R. M., *Stable Adaptive Control and Recursive Identification of Nonlinear Systems Using Radial Gaussian Networks*, PhD Thesis, MIT Dept. of Aeronautics and Astronautics, May 1993.
- [17] Sanner, R. M., and Slotine, J.-J. E., "Gaussian networks for direct adaptive control", *IEEE Trans. Neural Networks*, 3(6), pp 837-863, 1992.
- [18] Sanner, R. M., and Slotine, J.-J. E., "Stable adaptive control of robot manipulators using "neural" networks", to appear *Neural Computation*, 1995.
- [19] Sanner, R. M., and Vance, E. E., "Adaptive control of free-floating space robots using "neural" networks", submitted, 1995 *American Control Conference*, Sept 1994.
- [20] Sanner, R. M., and Vance, E. E., *SSL Report*, University of Maryland, Dept. of Aerospace Engineering, 1994 (in preparation).
- [21] Slotine, J.-J. E., and Di Benedetto, "Hamiltonian adaptive control of spacecraft", *IEEE Trans. Automatic Control*, 35, pp. 848-852, 1990.
- [22] Slotine, J.-J. E. and Li, W., *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [23] Slotine, J.-J. E., and Li, W., "On the adaptive control of robotic manipulators", *Intl. Journal of Rob. Res.*, 6(3), 1987.
- [24] Wen, J. T.-Y., and Kreutz-Delgado, K, "The attitude control problem", *IEEE Trans Autom. Control*, 36(10), 1148-1162, 1991.
- [25] Wie, B., Weiss, H., and Arapostathis, A., "Quaternion feedback regulator for spacecraft eigenaxis rotations", *J. Guidance, Control, and Dynamics*, 12(3), 375-380, 1989.
- [26] Wie, B., and Barba, P.M., "Quaternion feedback for spacecraft large angle maneuvers", *J. Guidance, Control, and Dynamics*, 8(3), 360-365, 1985.

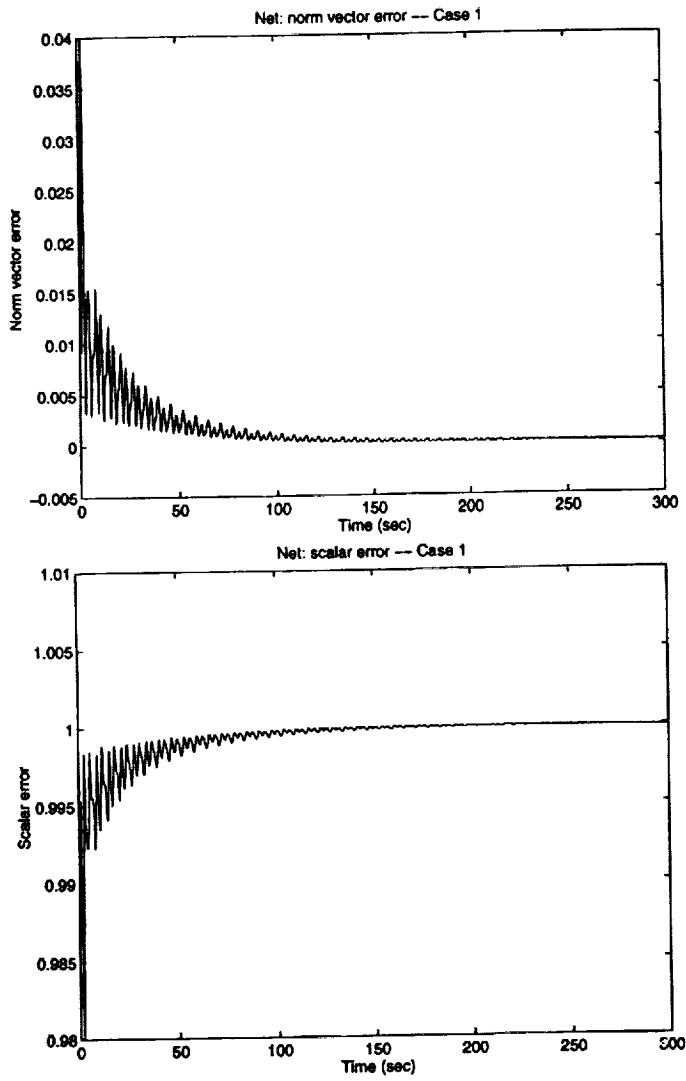


Figure 1: Attitude tracking performance using the proposed adaptive neurocontroller with the dynamics (1). The top figure shows the norm of the the vector part of the error quaternion, $\|\tilde{\epsilon}\|^2$, while the bottom figure shows the scalar part of the error quaternion, $\tilde{\eta}$.

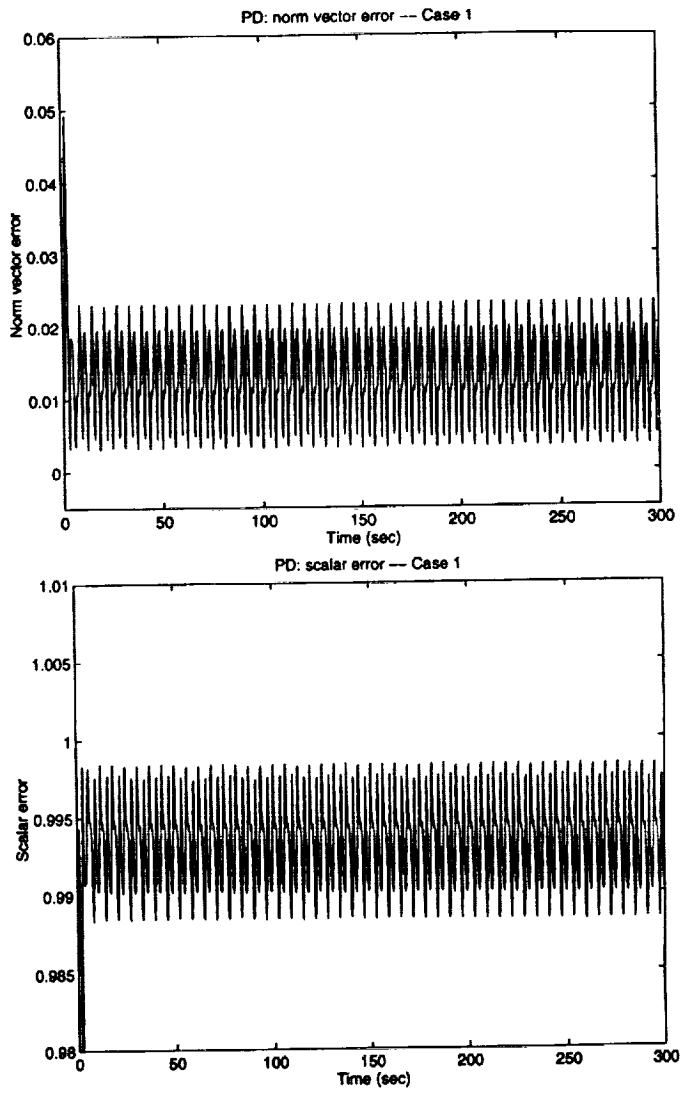


Figure 2: Attitude tracking performance with the dynamics (1) omitting the adaptive contribution of the neural network. The top figures shows the norm of the the vector part of the error quaternion, $\|\tilde{\epsilon}\|^2$ while the bottom figure shows the scalar part of the error quaternion, $\tilde{\eta}$.

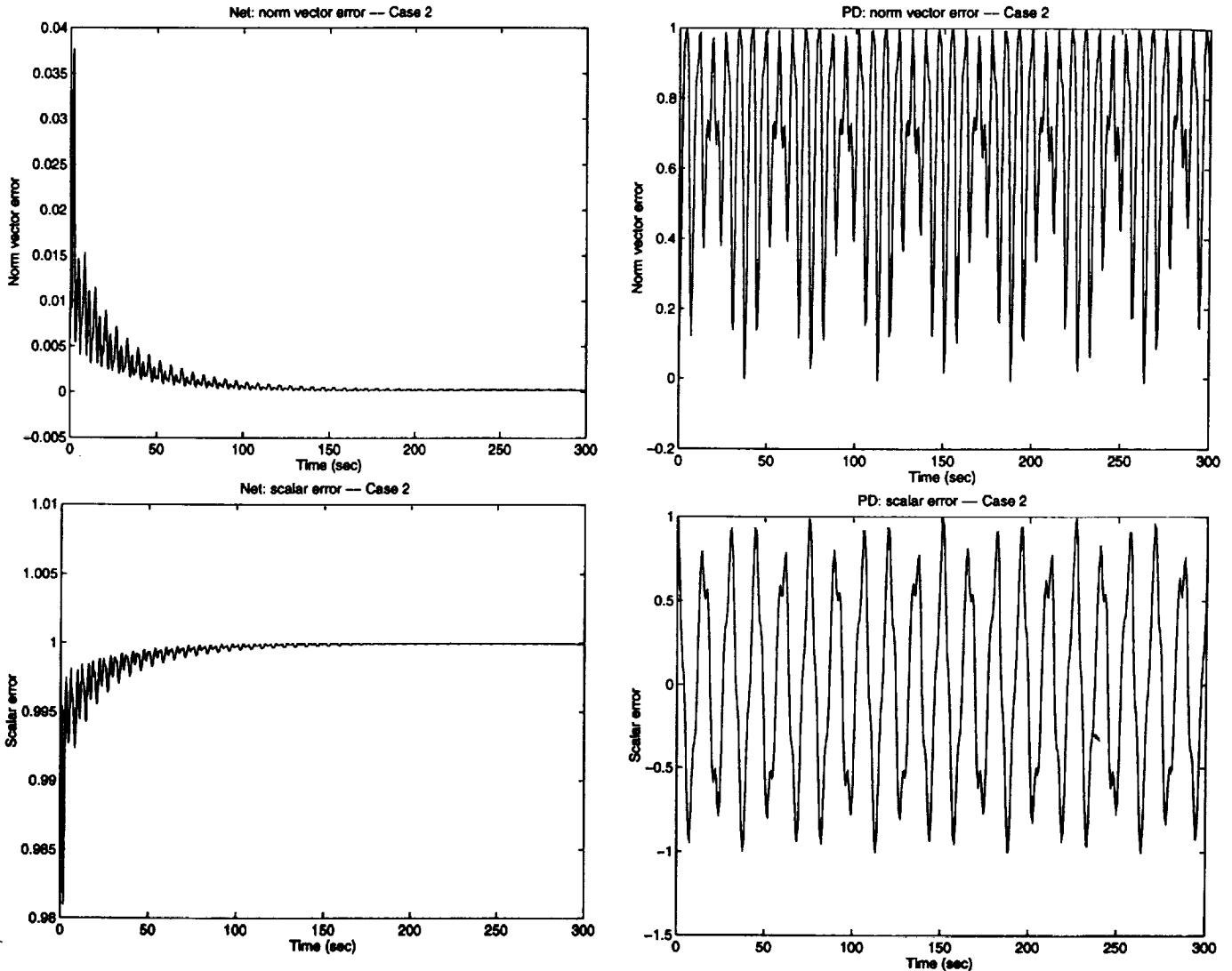


Figure 3: Attitude tracking performance using the proposed adaptive neurocontroller with the dynamics (14). The top figure shows the norm of the the vector part of the error quaternion, $\|\tilde{\epsilon}\|^2$, while the bottom figure shows the scalar part of the error quaternion, $\tilde{\eta}$.

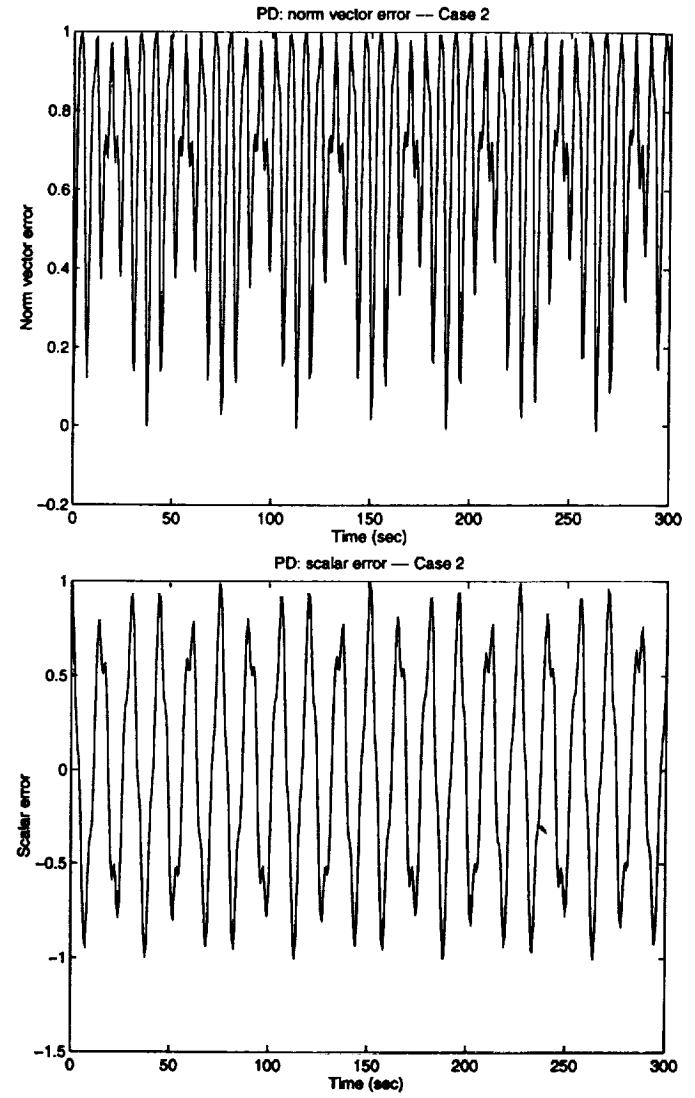


Figure 4: Attitude tracking performance with the dynamics (14) omitting the adaptive contribution of the neural network. The top figures shows the norm of the the vector part of the error quaternion, $\|\tilde{\epsilon}\|^2$ while the bottom figure shows the scalar part of the error quaternion, $\tilde{\eta}$.

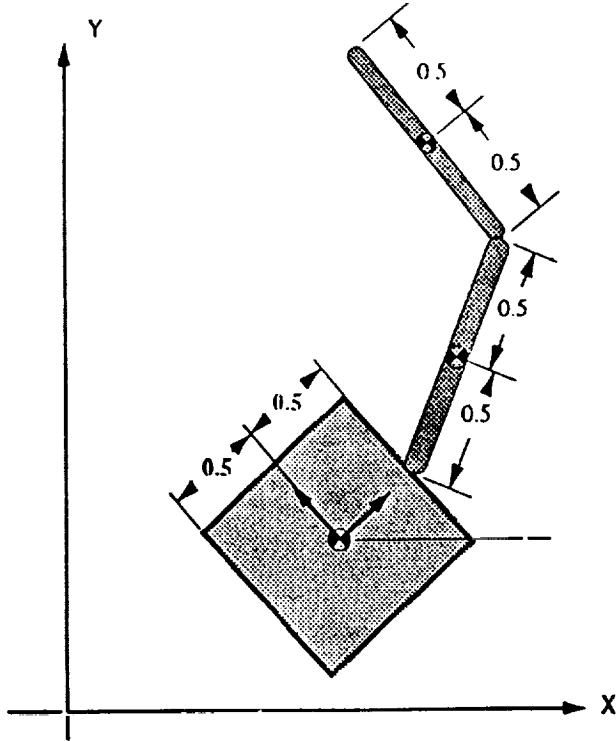


Figure 5: Diagram of the 3DOF simulation model

	spacecraft	link 1	link 2
mass (kg)	40	4	3
inertia ($\text{kg}\cdot\text{m}^2$)	6.667	.333	.250
length (m)	1	1	1

Figure 6: Physical parameters of the simulation model

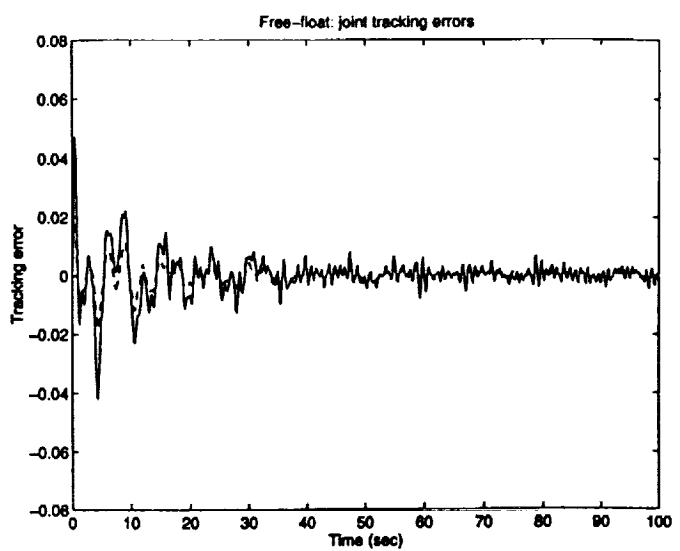


Figure 7: Joint angle tracking performance for the free-floating space robot using the proposed adaptive neurocontroller.

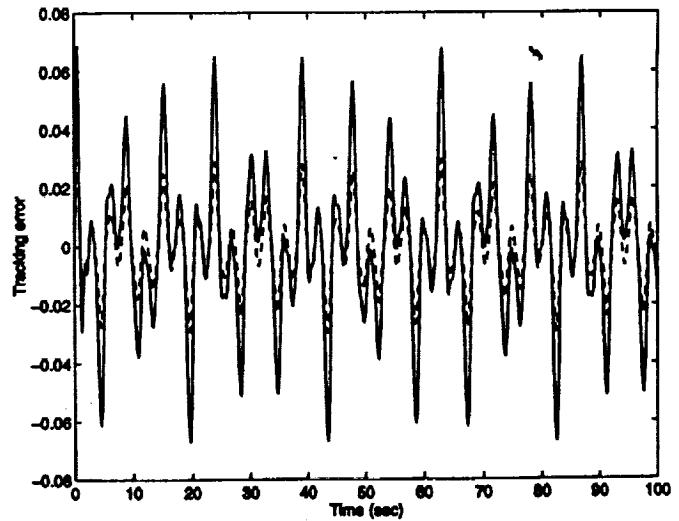


Figure 8: Joint angle tracking performance without use of the adaptive contribution of the neural network.

Fuzzy Logic Techniques for Rendezvous and Docking of Two Geostationary Satellites

Guillermo Ortega
European Space Agency
European Space Operations Centre
D-64276 Darmstadt, Germany
e-mail: g.ortega@ieee.org

S11-63

47261

p. 10

ABSTRACT

Large assemblies in space require the ability to manage Rendezvous and Docking operations. In future these techniques will be required for the gradual build up of big telecommunication platforms in the geostationary orbit.

The paper discusses the use of Fuzzy Logic to model and implement a control system for the docking/berthing of two satellites in geostationary orbit. The system mounted in a chaser vehicle determines the actual state of both satellites and generates torques to execute maneuvers to establish the structural latching. The paper describes the proximity operations to collocate the two satellites in the same orbital window, the Fuzzy guidance and navigation of the chaser approaching the target and the final Fuzzy berthing.

The Fuzzy Logic system represents a Knowledge Based Controller that realizes the close loop operations autonomously replacing the conventional control algorithms. The goal is to produce smooth control actions in the proximity of the target and during the docking to avoid disturbance torques in the final assembly orbit.

The knowledge of the Fuzzy controller consists of a data base of rules and the definitions of the fuzzy sets. The knowledge of an experienced spacecraft controller is captured into a set of rules forming the Rules Data Base.

INTRODUCTION

Since several years ago the number of spacecrafts in the geostationary orbit has increased considerably. Large satellites with longer life times are now the trends in the market. A big part of this development is due to commercial telecommunications companies with a growth on circuits demand of approximate 10% per year.

For the time being the satisfaction of this demand is fulfilled by a progressive tightening of the East-West deadband and the construction of complex clusters

with collision avoidance strategies. But the concept of satellite cluster is only attractive if the functioning of the set of spacecrafts in the same orbital window is seen by the user as only one payload.

Current practice for orbital windows in GEO assumes a square of about $\pm 0.1^\circ$ in latitude and longitude but this is gradually being shrunk to $\pm 0.05^\circ$. This demand in orbital space has a big impact on spacecraft design, ground station design and station keeping operations; the spacecraft must incorporate more efficient propulsion systems and satellite-to-satellite tracking devices (ESA 10035), the ground station has to increase in complexity to allow higher accuracy in

orbit determination and operations have to become more complex for inclination and eccentricity maintenance. Some solutions arise: satellite clusters (with heavy workload in ground operations or autonomous station keeping) or the gradual assembling of big platforms in orbit. The first technique requires orbit determination methods on board the satellite (where ground operations workload is considerably reduced) or on ground. The second requires the ability to master rendezvous and docking operations but once the assembly is formed the control remains purely conventional.

The advantages of a big platform formed by the joining of several pieces instead of a "dancing" cluster of satellites can be listed as follows:

1. reduction to zero the risk of collisions.
2. reduction in ground station workload.
3. possibility to add as many pieces as desired increasing by far the capacity of the initial payload.

On the other hand the satellites must to be constructed with docking mechanisms and autonomous control systems.

RENDEZVOUS MISSION DESIGN

Among the main three techniques used for a chaser approaching a target (R-bar, V-bar, inertial) the V-bar is the most common used (Leonard 89) for its stability properties. With this technique less propellant than R-bar is required but plume impingement problems appear. Here only V-bar option will be considered. For a mission of this type the following assumptions are made:

- ***the target:*** it is a geostationary communications satellite. The target is passive and has the necessary mechanical-electrical elements for the docking of the chaser. It is maintained in the centre of its orbital window within a specified inclination and eccentricity. The target is three-axis stabilized via a double-gimballed, bias momentum control system during the complete rendezvous mission.
- ***the chaser:*** it is another geostationary communications satellite with identical shape and mass. The chaser has a control system that allows the rendezvous and soft docking with the target in an autonomous form (no man-in-the-loop capabilities

are considered). If the rendezvous fails the chaser returns to a safety position. The chaser is approaching the target using the V-bar technique where the docking axis is along the velocity vector.

- ***the environment:*** the chosen reference coordinate system is the Local Vertical, Local Horizontal (LVLH) (fig. 1). That is: +X in the direction of target flight, +Z in the direction of center of Earth and +Y towards the Earth south pole. The orbital

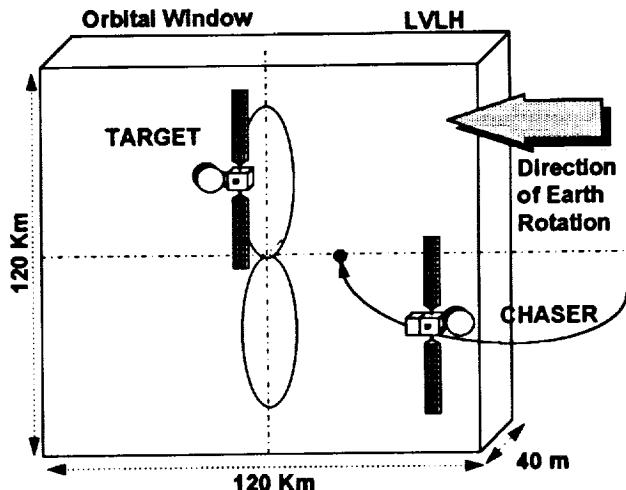


Figure 1. Final approach and injection into orbital window

window measures are $\pm 0.1^\circ$ in longitude and latitude. This window is maintained by means of ground operations with a full antenna coverage in all occasions. The control centre could be located in ESOC (Germany) having an antenna which monitors the first part of the operation.

Under these assumptions the type of mission design for an assembling in GEO of two satellites will not be different from the conventional design during the launch, transfer and drift orbits and the station acquisition phase (Pocha 87). The difference resides in the guidance and navigation of the chaser approaching the target and the final docking.

Launch and GTO Orbit

The launch can be performed from Kourou Space Centre in French Guyana using an Ariane 4 expendable three-stage rocket. Using this rocket the inclination of the transfer orbit is 7° . At the apogee of the transfer orbit the apogee motor is fired acquiring a

near geosynchronous orbit (the drift orbit).

Drift Orbit and Station Acquisition

During the drift orbit the chaser acquires its operational three-axis stabilized attitude pointing involving de-spin, Sun acquisition and Earth acquisition. For this orbit a set of 4 ESA ground stations to track the target can be assumed.

Once the satellite has acquired its operational attitude stabilization it must be placed closed to the target orbital longitude and with a specific orbit inclination. This operation is accomplished using ground tracking support. In this case only one station is involved.

Rendezvous & Docking Operations

These are defined as the set of operations to close up and dock two spacecrafts. In a typical mission they

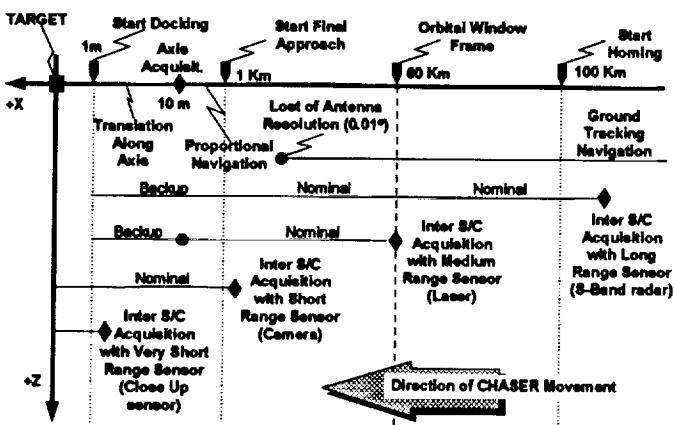


Figure 2. Rendezvous sequence timeline

comprise (fig 2): homing, final approach, docking and structural latching (Wohlke 92, Pairot 92).

Homing

This phase starts with the target presence acquisition by the large range sensor (S-band radar) mounted in the chaser. That happens approximately at -100 Km (LVLH system) behind and slightly below the target. In this phase although the ground station supplies navigation data, the radar works in back-up mode giving R, Az and El of the target. During this phase only translation movement of both target and chaser is considered.

At -60 Km the chaser crosses the orbital window and

in that moment the medium range sensor (laser) locks-on the target. During this part the laser sensor and radar sensor are nominal. Now the attitude pointing of the chaser starts to be important.

The homing phase extends to a distance of -1 Km (behind the target). In that moment another phase starts.

Final Approach

This phase comprises the close up of the chaser from -1 Km to -1 m. During this time the short range sensor (camera) localizes a specific mark in the target (Ho 93). In this phase the camera functions in nominal mode whereas the laser and the radar are in backup mode.

Docking

This phase starts at -1 m from the target and ends just a few centimeters from it, before the latching. Four close up sensors mounted in a cruciform way in the chaser west platform side allow the fine docking.

Latching

Four latches mounted clockwise to the close up sensors will fit into four handles that close when the proximity operations are finished (Fehse 85). In this phase the sensors and the camera operate in parallel.

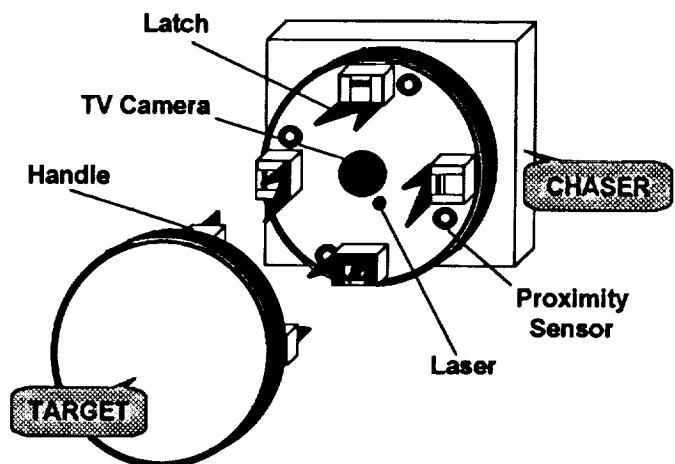


Figure 3. Docking mechanism

The laser and radar data are not considered (fig. 3).

THE PHYSICS OF THE SYSTEM

The translational motion of a spacecraft system in circular orbit can be described using the Clohessy-Wiltshire equations. Those are linear differential equations with time constant coefficients which describe the movement of two masses in a circular orbit around a third object. They were programmed in the rendezvous guidance computer used in the Gemini mission (1962) and still provide short-range maneuver computation for the Shuttle. Nowadays the range of validity of these equations have been extended by introducing special coelliptic coordinates to generalize the LVLH reference system.

The Clohessy-Wiltshire equations have as input the initial position and velocity of the chaser. The output is the position and velocity of the chaser after a time interval. To apply these equations the two rotating bodies must have a small mass in comparison to the non-rotating body. In addition the target is located in the origin of the rotating coordinate system. The position and velocity of the chaser are given in this LVLH reference system.

$$\ddot{\mathbf{r}}_2 - \ddot{\mathbf{r}}_1 = \frac{\mu}{r_1^3} \left[\mathbf{r}_1 - \frac{r_1^3}{r_2^3} \mathbf{r}_2 \right] + \mathbf{f}$$

where r_1, r_2 are the distances of target and chaser from the Earth centre, μ is the product $G*M_{\text{earth}}$ and \mathbf{f} is the perturbing force. This equation linearised and written in terms of Cartesian coordinates (LVLH system) gives:

$$x - 2\omega z = f_x$$

$$y + \omega^2 z = f_y$$

$$z + 2\omega x - 3\omega^2 z = f_z$$

These equations are not solvable in general but in some special cases it is possible to derive an analytical solution. If the external forces are zero ($\mathbf{f} = 0$) the equations can be applied to a rendezvous in high LEO orbits (Shuttle, Soyuz-MIR) giving excellent results. (Malyshev 94, Brown 94). For a rendezvous in low LEO the force \mathbf{f} is the atmospheric drag, etc.

In the geostationary case several forces (apart from the Earth gravity) perturb the orbit of satellites: the Moon, Sun, Earth triaxiality, solar wind, etc. The Moon-Sun combined effect causes the orbit inclina-

tion to grow $0.85^\circ \text{yr}^{-1}$ in average. The Earth's oblateness causes a precession of the ascending node of about 4.9°yr^{-1} . This effect is noticeable for non inclined orbits but it is negligible for orbits with 0° inclination. The Earth triaxiality causes a longitudinal acceleration towards GEO points at 79°E and 107.6°W . An average value in acceleration of 0.001° per day² towards these points is typical. Finally the solar radiation pressure varies as the inverse square of the distance from the Sun. This force depends on the type of spacecraft surface and can be estimated as:

$$F_{sp} = k * A * 1.6 * \cos i$$

The magnitude of this force along pitch axis is $10^{-4} \cdot \cos \omega_{\text{geo}} \cdot t \text{ Nm}$ and the magnitude along roll and yaw axis is $10^{-5} \cdot \cos \omega_{\text{geo}} \cdot t \text{ Nm}$.

Once the chaser is close to the target during the final approach phase the force of the Earth gravity can be considered uniform. In this case the reference system becomes inertial body reference system and the equations become:

$$m_{\text{chaser}} x = f_x$$

$$m_{\text{chaser}} y = f_y$$

$$m_{\text{chaser}} z = f_z$$

where again \mathbf{f} is the perturbing force and x, y, z are

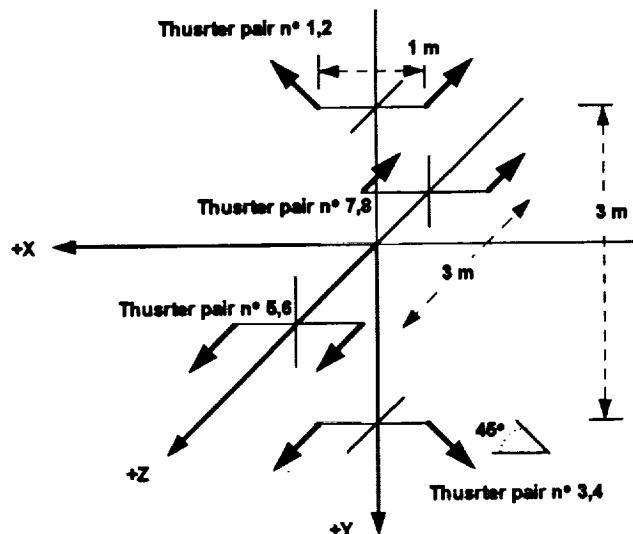


Figure 4. Thrust location in chaser

the chaser coordinates with respect to the LVLH centre. The target system physics of movement can be described using body coordinate equations as well.

The translation of both vehicles is realized by means of thrust impulses. The thruster system consists of 8 pairs localized at four surfaces as shown in (fig. 4). Pairs (5,6) will impulse the satellite in the -X direction and pairs (7,8) in +X direction, etc.

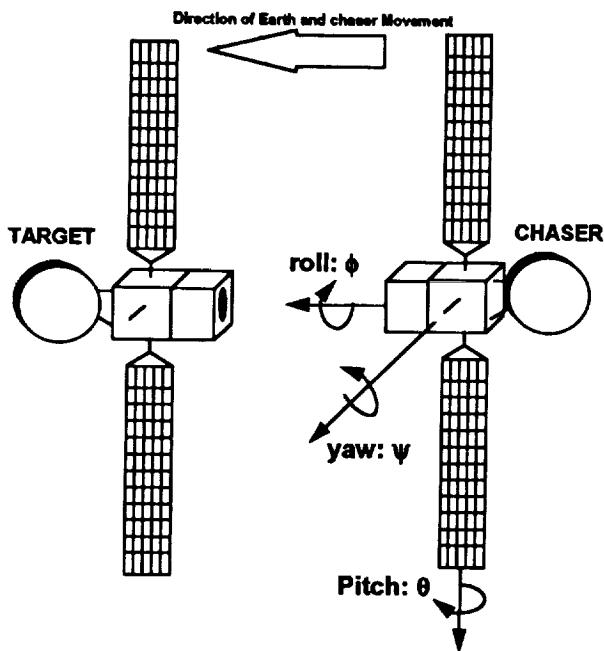


Figure 5. Final closing body coordinate system

The attitude control of both chaser and target is designed as a three axis stabilized, double gimballed, bias momentum control system (Kaplan 76, Wertz 78). The corresponding Euler equation is

$$(\mathbf{F}_d + \mathbf{F}_{ss})|_{S'} = \left[\frac{d\mathbf{h}}{dt} \right]_S + \mathbf{w}_{ss} \times \mathbf{h}_s$$

where S' is a fixed reference system with origin in the centre of satellite mass and S is a reference system with the same origin that rotates with the satellite. S' and S are defined as the LVLH system. \mathbf{F}_d is the disturbance force (solar pressure, thruster misalignment, etc.) and \mathbf{F}_{ss} is the force due to gravity gradient. \mathbf{w}_{ss} is the angular velocity of S respect to S' and \mathbf{h} is the total angular momentum of the spacecraft (body + gimballed wheel).

Pitch equation is simple and decoupled from yaw and roll assuming a symmetric satellite

$$F_{\phi} = I_x \ddot{\theta} + h_z$$

but yaw and roll equations are coupled:

$$F_{\dot{x}} = I_x \ddot{\phi} + \omega_{ss} h_y \phi + h_y \dot{\psi} + h_x - \omega_{ss} h_z$$

$$F_{\dot{y}} = I_y \ddot{\psi} + \omega_{ss} h_x \psi - h_x \dot{\phi} + h_z - \omega_{ss} h_x$$

where I_x , I_y , I_z are the principal moments of inertia and the control torques (h_x , h_y , h_z) are produced through gimbals deflections (fig. 5).

FUZZY CONTROL: SYSTEM MODEL

Figure 6 shows the proposed Guidance, Navigation and Control System (GNC). The navigation block calculates the actual state of both spacecrafts and their relative measurements. The guidance and control blocks are together in the figure. The guidance part calculates the future state of the spacecraft to achieve

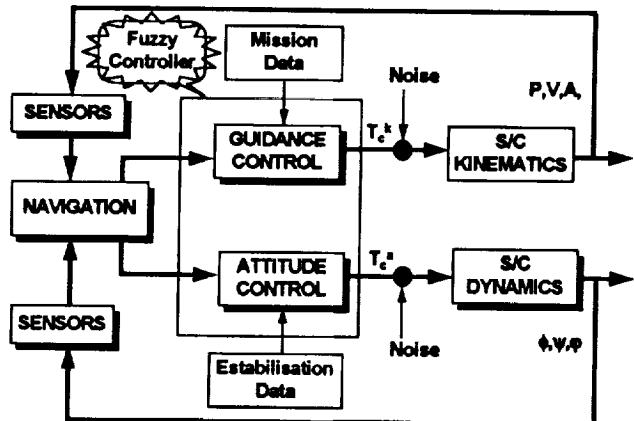


Figure 6. Guidance, navigation and control system

the desired trajectory and the control part calculates the desired control torques to achieve this trajectory.

It must be noticed that the guidance and control (G&C) system is divided in two parts: the G&C of the translation movement and the G&C of the rotational movement. For this study both movements are supposed to be de-coupled and the influence of one on the other is considered negligible. However we will see that the attitude motion influence on the G&C

part of the translation motion.

Controller Implementation

The fuzzy controller (FC) must substitute the guidance and control parts. The goal of the FC is to be able to generate the control signals derived from the sensor measurements (Lee 90). The steps involved in the construction of this FC are the following:

1. define input and output variables.
2. define universe of discourse for all variables.
3. define the Rule Data Base (RDB).
4. define the Inference Engine

Once the FC is constructed it must be coded and included in the simulation.

For all the four previous points a fundamental step in the implementation of the FC is to capture the knowledge of an expert. This knowledge must be applied to guide and to control the vehicle. The expert will help defining the way to determine the state of the target-chaser system and the way to use the commands for each situation. The inference engine can be an approximate reasoning kernel based on already proposed systems (Buckley 92, King 94).

Construction of IFS and OFS

In this case the input variables are the sensor measurements (positions, velocities, etc.) and the output variables are the firing of thrust (thrust position and time of fire) and the attitude angles and rates. The input variables will be represented in the Input Fuzzy Sets (IFS) system and the output variables will be represented in the Output Fuzzy Set (OFS) system (Drianov 93).

For the translation movement it is simpler to use input variables in polar coordinates rather than Cartesian: an aircraft pilot is able to measure azimuth and elevation angles and rates and distances and distances rates. The rotational movement can be represented by the attitude angles and angles rates. The output for a human controller could be the deflection of a joystick to fire a particular thruster and the deflection time for acceleration controlled devices or the amount of pulses in pulse control devices.

Defining the control variables

For the translation movement of the chaser the input variables of the FC system will be azimuth (a), eleva-

tion (e), azimut rate (\dot{a}), elevation rate (\dot{e}), range (r) and range rate (\dot{r}). For the rotational part the input variables will be pitch (θ), pitch rate ($\dot{\theta}$), roll (ϕ), roll rate ($\dot{\phi}$), yaw (ψ) and yaw rate ($\dot{\psi}$). The output variables will be the amount of firing time (ft) and the position of fired thruster (pt).

The universes of discourse of these variables are as follows:

$$a \in [-\pi, \pi] \text{ in rad}, \dot{a} \in [-2, 2] \text{ in } ^\circ/\text{sec}$$

$$e \in [-\pi, \pi] \text{ in rad}, \dot{e} \in [-2, 2] \text{ in } ^\circ/\text{sec}$$

$$r \in [0, R_{\max}] \text{ in Km}, \dot{r} \in [-10, 10] \text{ in m/s}$$

$$\theta \in [-\pi, \pi] \text{ in rad}, \dot{\theta} \in [-0.5, 0.5] \text{ in } ^\circ/\text{sec}$$

$$\phi \in [-\pi, \pi] \text{ in rad}, \dot{\phi} \in [-0.5, 0.5] \text{ in } ^\circ/\text{sec}$$

$$\psi \in [-\pi, \pi] \text{ in rad}, \dot{\psi} \in [-0.5, 0.5] \text{ in } ^\circ/\text{sec}$$

$$\text{Burn} \in [-10, 10] \text{ in dcm/sec}$$

Defining the Fuzzy Sets

The fuzzy subdivisions of the universe of discourse of any variable are called the fuzzy sets. Every 'partial'

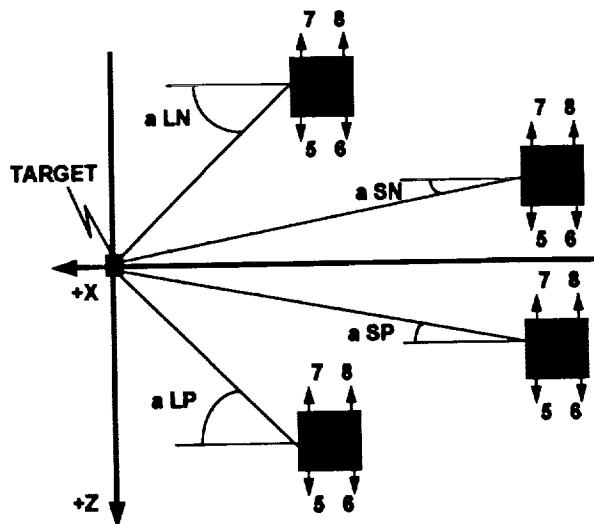


Figure 7. Azimut fuzzy set selection

value of each variable can belong to more than one set.

For this study the following sets are assumed:

- * for azimuth, azimuth rate, elevation, elevation rate and range rate the fuzzy sets are *Small Negative* (SN), *Small Positive* (SP), *Large Negative* (LN) and *Large Positive* (LP) (fig. 7).
- * for range the possible fuzzy sets are *Small* (S) and *Large* (L) as no more distinctions are needed.
- * for roll, pitch, yaw and its derivatives the fuzzy sets are *Small Negative* (SN), *Small Positive* (SP), *Large Negative* (LN) and *Large Positive* (LP)

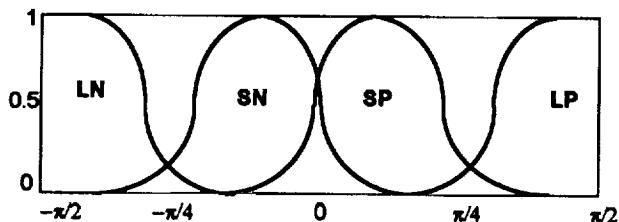


Figure 8. Fuzzy representation for azimuth

Special attention must be put when creating the size and type for the fuzzy membership functions. Depending on the position along the universe of discourse axis, the control actions will be more or less smooth. A rule for relative good control states that the functions should overlap 25%. The profile of the function can be trapezoid, bell shaped, gaussian, etc. A first approach in the definition of the membership functions for the problem of the rendezvous is shown in figure 8.

The azimuth of the chaser (movement along X axis) is subdivided in the four fuzzy sets (*LN*, *SN*, *SP*, *LP*). For small angles ($\alpha \in [-\pi/4, \pi/4]$) the exponential gaussian shapes overlap a little bit more than 25 % trying to smooth thruster actions within these limits.

Rules Data Base

The rules data base is the kernel of the knowledge base controller (KBC). Thanks to this data base the FC will incorporate an experience which can only be realized in the corresponding analytic model by means of manual operations. In this case the KBC implements the close loop control actions substituting the operator (Tolk 94).

Capturing the knowledge

Here the expert tells us the following:

⇒ the control of each axis is carried out in an independent manner: firings in Z direction control azimuth and firings in Y direction control elevation. The firings over each axis are calculated taking into account angles and its derivatives.

⇒ the control in azimuth and elevation is not symmetric: due to the Clohessy-Wiltshire equations of movement firings in X direction will 'elevate' the chaser in its path towards the target. It is necessary to realize compensation firings in the +Z direction. However, the control in elevation is symmetric with respect to the +X axis.

⇒ the tendency of human controllers is to make azimuth and elevation equal to 0° during the final approach phase (typically at -10 or -20 m of the target) to be able to reduce control workload and focus on fine range rate control. This scheme complies with an intuitive proportional navigation guidance towards the point of starting axis translation.

⇒ the amount and size of the firings depend proportionally on the distance: different firing strategies must be followed depending on the distance to the target. When the chaser is in homing phase the amount of firings is low (typically 1 every 5 minutes) and the size is big. In final approach phase the firings amount starts to increase (1 every minute) but the size gets small. During docking the firings are frequent with very small size.

⇒ attitude angles and angles rates are controlled during the translation along axis phase.

Constructing the knowledge data base

The rules data base are given in the following form:

if ANTECEDENT then CONSEQUENT

where ANTECEDENT and CONSEQUENT are any composition of statements (and, or, not, etc). The statements contain declarations of associations of fuzzy variables to fuzzy sets (fig. 9).

These rules are grouped in two categories and every category contains several groups of control. One category corresponds to a long distance between target and chaser and the other corresponds to short distance between target and chaser. This distinction allows to fire different sets of rules belonging to translation or rotational movement: during long distances (r is large) no rule of rotational movement is fired. When the chaser is getting closer to the target

the relative attitude of both vehicles starts to be im-

crafts are simulated using the Euler equations with gravity gradient and solar pressure perturbations.

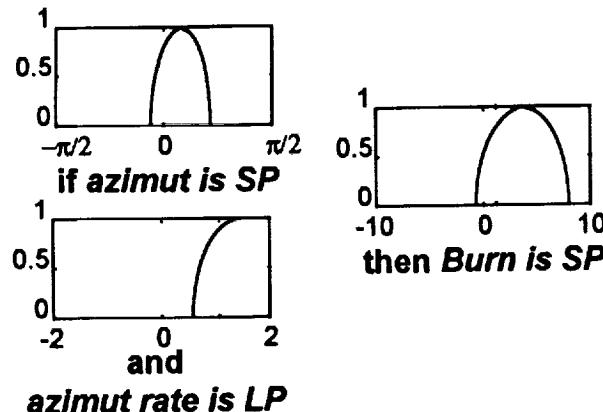


Figure 9. Azimut rule

portant.

For each category the rules are grouped depending on the control action they generate: for the translational movement there are rules for controlling azimut, rules for controlling elevation and rules for controlling distance; for the rotational movement there are rules for controlling pitch, rules for controlling roll and rules for controlling yaw;

SIMULATIONS

For the simulations, the size of the satellites is assumed to be about 15 m high from the extreme ends of the solar panels and the body size is about 3 m³ (e.g. ESA ECS satellite). The mass is 1000 Kgr. The matrix of moments of inertia is diagonal. The thruster misalignment torque is of an order of magnitude less than the solar pressure. The attitude pointing requirements for the soft docking are 0.005° in pitch and roll and 0.4° in yaw. The docking velocity is 1 cm/s.

The plant description is simulated using the Clohessy-Wiltshire equations for the chaser influenced by a constant force in the -Y direction (Moon-Sun attraction) during the first part of the rendezvous. During the final approach phase the reference system is inertial body and the equations get linear. The target translation is simulated with a constant force in the same direction. The attitude dynamics of both space-

The FC controller is implemented in the following way:

- The control variables together with their universes of discourse are as described before.
- The shape of the membership functions is an exponential gaussian function.
- The rule data base is composed of 32 rules (20 for translation and 12 for rotation).
- The inference engine is programmed using the Mamdani's Min-Max mechanism (Mamdani 74); the AND operator is chosen as the minimum of two weight antecedents instead of its multiplication.
- The defuzzification strategy used is the centre of gravity computation.

The pseudocode of the simulation is as follows:

```

initialize all;
create rules for the rules data base;
step = 1;
big_loop
  for t = 1 to time to fire again
    draw positions & velocities;
    chaser kinematics;
    target kinematics;
    chaser attitude dynamics;
    target attitude dynamics;
    sensor measurements;
    store everything for drawing;
  end for;
  Fuzzy_control_computations;
  if R = 0 then out
  else step = step+1;
  end big_loop;

```

Figure 10 shows a simulation run. The chaser was initially located in the coordinates (-1,0,1) behind the target at -1Km. The firing of thrusters try to close the chaser to the docking axis as soon as possible. This is practically achieved at -400 m. The translation along the axis occurred at -10 m. The azimut of the chaser decayed from 45° to 10° in a nearly exponential curve. From 10° to 0° the azimut waved up and down. This movement corresponds to the different trials to align with the docking axis.

Figures 11a and 11b show another simulation run. This time the chaser was initially located in the coor-

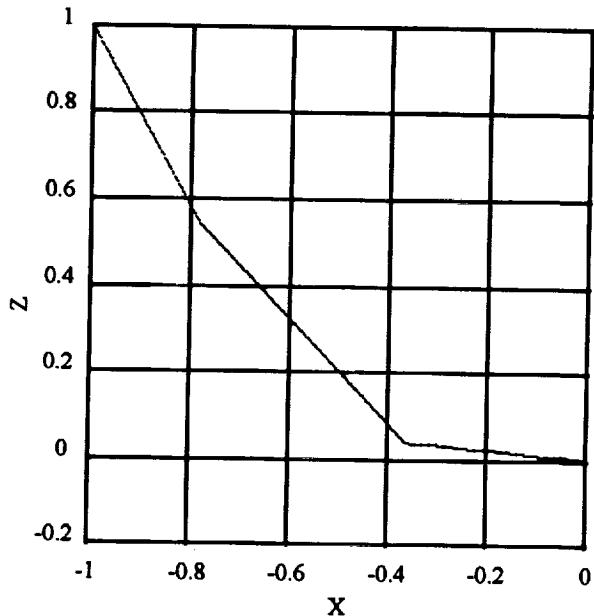


Figure 10. Simulation run (in-plane motion)

dinates (-0.05, -0.05, 0.05) behind the target at -500 m and below the target at -500 m as well. The firing of thrusters, for the in-plane motion (fig. 11a), try to

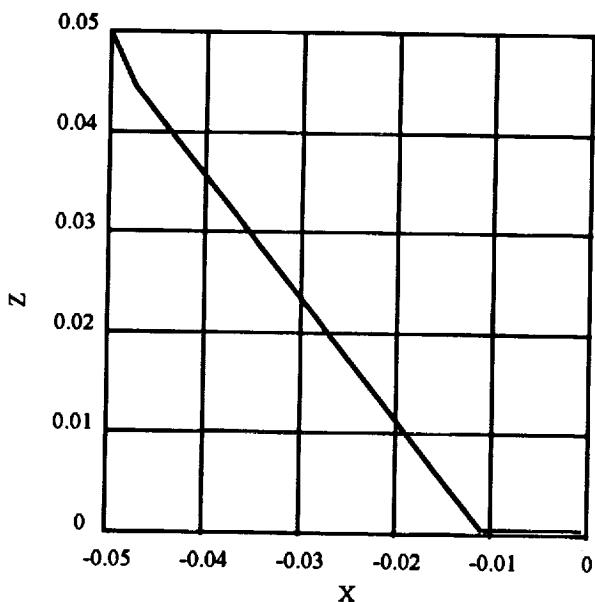


Figure 11a. Simulation run (azimuth) in-plane motion

close the chaser to the docking axis as soon as possible. For the out-of-plane motion (fig. 11b) the firings were almost linear (because the angular velocity of

the orbit is very low). The translation along the axis occurred at -11 m. The azimuth of the chaser decayed again from 45° to 10° in a nearly exponential curve. From 10° to 0° the azimuth waved up and down as well. The elevation of the chaser decayed from 45° to 0° in a nearly linear curve. From 10° to 0° the elevation did not wave up and down.

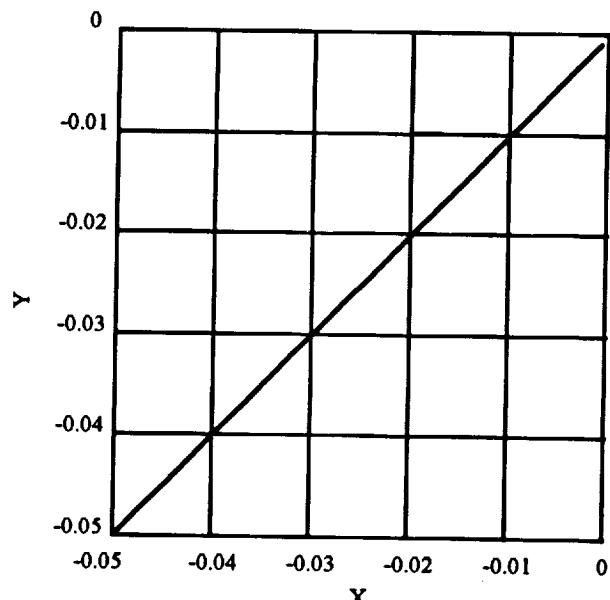


Figure 11b. Simulation run (elevation) out-of-plane motion

Simulations were developed on a PC compatible 486/66DX2 portable type computer. The code was written in MATLAB 4.0® for Microsoft Windows™. The fuzzy inference engine with approximate reasoning was implemented using FISMAT, the Fuzzy Inference Systems toolbox for MATLAB developed by A. Lofti. FISMAT provided fuzzy logic operations and different methods of approximate reasoning.

CONTINGENCIES

Contingency situations can happen when the docking velocity is too high, the docking axis is not in perfect alignment with the V-bar axis or when the attitude of both satellites differ in an angle bigger than the one specified for the latching mechanism.

Easily, the FC can be extended to handle these situations. For an hazardous impact the thrust pair (1,3) must be used to produce an exponential breaking

during the translation along axis. This situation was taken into account with the rule

if range rate is L then Burn (1,3) is SN

For the situation with the docking axis not in perfect alignment with the V-bar axis a pseudo proportional navigation towards the starting point of the axis translation was implemented with the rule

if a is LP and a rate is LP then Burn (5,6) is LP

Finally when attitude angles are out of range of the specifications of the latching mechanism some corrective measures can be taken during the translation along-axis phase.

CONCLUSIONS

Up to date, servicing is greatly dependent of human operations in space. Therefore, the on-orbit accessibility factors of the satellite to be serviced is a primary consideration. Support capabilities for human presence in satellite maintenance tasks in GEO will not be widely available until the first decade of the 21st century (Waltz 90). In the meantime autonomous rendezvous and docking arise as a good alternative.

Fuzzy logic emulates the behavior of human operators for complex control tasks. A fuzzy logic controller embedded in a guidance, navigation and control system of a spacecraft can realize autonomously the close loop operations replacing the conventional crisp control algorithms. The fuzzy controller produces soft control actions in the proximity of the target and during the docking to avoid disturbance torques in the final assembly.

Fuzzy controllers can be programmed, tested and qualified for flight (Daley 85, 87). Its rule data base can be constructed with the help of an expert and refined in simulations. Fuzzy controllers are easily reconfigurable for different type of missions and their performance is robust to changes.

REFERENCES

- Study on Intersatellite Tracking for Co-location (ESA/10035/92/D/CS)
- An Introduction to Mission Design for Geostationary Satellites (J.J. Pocha; D. Reidel Publishing Company 1987, ISBN 90-277-2497-2).
- Rendezvous and Berthing between Columbus Free Flying Laboratory and Space Station Freedom (W. Wohlke; Automated Control in Aerospace 1992).

Development of an Autonomous Onboard Control System for Rendezvous and Docking (J.M. Pairot, W. Fehse, A. Getzschmann; Automated Control in Aerospace 1992).

Rendezvous and Docking Technology Development for Future European Missions (W. Fehse; ESA Journal 1985, vol. 9).

Modern Approaches on Guidance and Control of Russian Spacecraft (V.V. Malyshev, M.N. Krasilshikov; 2nd ESA International Conference on Spacecraft Guidance, Navigation and Control Systems, Aril.

Fuzzy Logic Application for Modeling Man-In-The-Loop Space Shuttle Proximity Operations (R. Brown, NASA-CR-188283).

Modern Spacecraft Dynamics and Control (M.H. Kaplan; John Wiley & Sons 1976, ISBN 0-471-45703-5).

Spacecraft Attitude Determination and Control (J.R. Wertz; Kluwer Academic Publishers 1978).

Fuzzy Logic in Control Systems Parts I and II (Chuen Chien Lee; IEEE Transaction on Systems, Man and Cybernetics, Vol. 20, No. 2, March-April 1990).

Universal Fuzzy Controller (J.J. Buckley; Automatica: Journal of the International Federation of Automatic Control -IFAC-, Vol. 28 Number 6, November 1992).

Multivariable Fuzzy Controller Design (R.E. King, G.D. Magoulas, A.A. Stathaki; Automatica: Journal of the International Federation of Automatic Control -IFAC-, Vol. 2 Number 3, June 1994).

An Introduction to Fuzzy Control (D. Drianov, H. Heldoorn, M. Reinfrank; Springer-Berlag 1993).

Fuzzy Logic: A Practical Approach (F.M. McNeill, E. Thro; Academic Press 1994, ISBN 0-12-485965-8 Automatic Programming of Fuzzy Controllers by Examples (A. Tolk, K. Helling; AI-Journal, June 1994).

Application of fuzzy algorithm for control of simple dynamic plant (E.H. Mamdami; Proc. IEE, 121, no. 12, December 1974).

A Justification for the Wider use of Fuzzy Logic Control Algorithms (S. Daley, K.F. Gill; Proceedings of the Institute of Mechanical Engineering, Vol. 199, No. C1, 1985).

Attitude Control of a Spacecraft Using an Extended Self-Organizing Fuzzy Logic Controller (S. Daley, K.F. Gill; Proceedings of the Institute of Mechanical Engineering, Vol. 201, No. C2, 1987)

Automatic Spacecraft Docking Using Computer Vision-Based Guidance and Control Techniques (Chi-Chang J. Ho, N. Harris McClamrock; Journal of Guidance, Control and Dynamics, vol. 16, No. 2, March-April 1993).

Disclaimer

The opinions expressed in this paper belong to its author and do not necessarily correspond to the opinions of ESA as a public institution.

amIT

Image Processing and Data Classification



A Hybrid Genetic Algorithm for Resolving Closely Spaced Objects

5/2-63

49262

J-8

R. J. Abbott

*The Aerospace Corporation
and
California State University,
Los Angeles*

W. E. Lillo

The Aerospace Corporation

N. Schulenburg

The Aerospace Corporation

A hybrid genetic algorithm is described for performing the difficult optimization task of resolving closely-spaced objects appearing in space-based and ground-based surveillance data. This application of genetic algorithms is unusual in that it uses a powerful domain-specific operation as a genetic operator. Results of applying the algorithm to real data from telescopic observations of a star field are presented.

1.0 Introduction

Extracting information on individual visual point sources in a closely-spaced object (CSO) cluster is a fundamental problem for such applications as astronomy and ballistic missile defense. The problem is difficult because objects within closely-spaced object clumps cannot be resolved directly. Instead, one hypothesizes overlapping point sources to create a model of the clump. Then one parametrically solves for the number of sources along with their amplitudes and locations.

An objective function is formed based on the sum of the squared residual errors between the data and model employed in Bayes' Theorem. This Bayesian approach has been described by [Sc93] and [Li94]. The best model is that which minimizes the residual errors and thus maxi-

mizes the probability that the model represents the data.

This estimation approach presents a difficult optimization task since the probability function to be maximized is rugged, i.e., has many local maxima.

Traditional approaches were found to be inadequate to solve this problem. Hill-climbing techniques were found to be highly dependent upon the initialization of the parameters; different solutions would be obtained from different initial guesses. Further, convergence was often slow, prohibitively so when the number of sources in the closely-spaced object clump reached four or more. To avoid these deficiencies, we developed a hybrid genetic algorithm.

The remainder of this paper is organized as follows. Section 2 describes the problem in slightly more detail. Section 3 provides an overview of genetic algorithms. Section 4 describes the genetic algorithm developed for the CSO problem. Section 5 presents our results.

2.0 Problem Statement

Every optical system has a point response function (PRF), which is the image generated by a sensor from a point source located at infinity. The PRF width is due to diffraction of the input radiation through the system aperture and the presence of aberrations in the optical system. Because their geometrical angular

subtense is much less than the width of the sensor PRF, many objects viewed by optical sensors, such as stars or distant space vehicles, appear as point sources.

If multiple point sources are located within the resolution limit of the sensor, the optical system will produce an image which appears as a *clump*. The objects in this case are referred to as a cluster of closely-spaced objects. The individual source amplitudes and locations determine the amount of overlap between the responses and thus the shape of the clump. Given the input clump of data and knowledge of the PRF, the sensor processing software must properly count and recover location and amplitude information of the individual objects.

3.0 Overview of Genetic Algorithms

The term *Genetic Algorithms* [Ho75] includes a broad class of iterative optimization techniques that employ methods that are modelled after the way evolution occurs by natural selection in biological systems.

3.1 The Genetic Algorithm

A genetic algorithm begins with a set of (sub-optimal) solutions, called a population. The initial population may be arbitrarily or randomly chosen, or it may be given as an external input.

An application-specific objective function is applied to each member of the population, thereby ranking the solutions.

The following selection/transformation/replacement cycle is repeated until a termination condition is met. (See Figure 1.)

1. **Selection.** Select one or more elements from the population using the following rule: the higher an element's score on the objective function, the *more* likely it is to be selected.

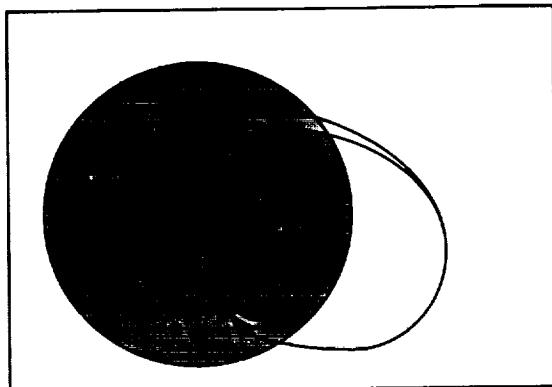


Figure 1. A generic genetic algorithm

2. **Transformation.** Modify the selected element or elements to produce a new, possibly higher ranked element. The operators used to modify the elements are called genetic operators.

- A single element may be modified in a process called *mutation*.
- Two or more elements may be combined to produce a new element. The process of combination can create new elements that combine the best attributes of their predecessors in ways that are very unlikely under purely random stochastic methods. This is widely considered as one of the sources of the efficiency and broad applicability of genetic algorithms.

3. **Replacement.** Put the new element back into the population, replacing some element currently in the population. The higher a population element's score on the objective function, the *less* likely it will be selected to be replaced.

When the process terminates, the best ranked solution is the reported solution.

Genetic algorithms have been successfully applied to a wide range of optimization problems including the travelling salesman problem [Gr85], communication network design [Da87], natural gas pipeline control [Go83],

image processing [Fi84], and training artificial neural networks [Wh89].

3.2 Hybrid Genetic Algorithms

As originally formulated, genetic algorithms were applied strictly to populations of fixed-length bit strings. All problem-specific information was encoded as bits. Within that framework, there are two genetic operators: mutation and crossover. The mutation operator changes one of the bits of the element to which it is applied. The crossover operator creates a new element by selecting, for each bit position, a bit in that position from the parents.

Hybrid genetic algorithms [Da91] move away from the bitstring representation in two ways.

1. Population elements are represented in ways that may be specific to the problem domain. Any data structure is allowed.
2. Genetic operators are defined which operate on the elements as represented. The primary operations are still generically mutation (change a single population element) and combination (combine pieces of multiple population elements to produce a new element). But mutation and combination are now tailored to the particular problem.

4.0 Application of Genetic Algorithms to the CSO Problem

In our application of hybrid genetic algorithms, we are searching for a set of objects at particular positions such that those objects would generate the given image. Since the number of objects to be resolved is unknown, the number of objects represented by each population element is not fixed. The positions of the objects are also unknown. The objective of the GA is to find both the number and placement of objects that best matches the received signal.

For the purposes of this discussion we need to distinguish terminologically between *elements in the population* and *objects to be resolved*.

We use *element* and *object* to make this terminological distinction. *Element* refers to an element of the population. *Object* refers to a signal generating object. A population *element* thus consists of a number of *objects*.

4.1 Element representation

We have simplified the problem in a number of ways.

1. Instead of attempting to find the location of the objects in 3-dimensional space, we limit the problem to finding positions of the objects in the 2-dimensional field of view of the sensor.
2. As a result of orthonormalization (see [Br87]), the brightness (or amplitude) of the hypothesized objects need not be considered during the search. The amplitudes are computed after a best solution is found. This is consistent with the principle of maximum entropy as explained in [Li94].
3. The PRF generated by each object is assumed known.

As a consequence of these simplifications, each object may be represented by its centroid, a pair of numbers, representing the $\langle x, y \rangle$ position of the hypothesized object in the two-dimensional field of view of the sensor.

Our population therefore consists of sets of elements, each of which is represented by its 2-dimensional centroid.

To simplify comparing one element with another, we order the objects in each element by their x-coordinate. Thus each population element is an ordered list of $\langle x, y \rangle$ pairs of numbers.

4.2 Smoothing the Ruggedness of the Search Space

The search space for this problem turns out to be very *rugged*. By this we mean two things.

1. A very small change in the $\langle x, y \rangle$ position of an object can make a very big difference in the value of the objective function of the element within which that object appears.
2. There are a great many local maxima.

The more rugged the search space, the more difficult the optimization problem. To make matters worse, we do not know in advance how many elements will best approximate the given signal. Thus the problem is not only to find the best solution in a rugged n-dimensional search space, it is also to find the best dimensionality.

The naive approach would be to let the genetic algorithm determine both the dimensionality and the position of the elements within that n-dimensional space. With no information, this approach would work, but it would be quite slow.

We did some experiments and found that we could “climb the dimensionality ladder” as follows.

- First run a genetic algorithm to determine the best 1-dimensional (i.e., one object) solution.
- Using that 1-dimensional solution to help seed the population, run the genetic algorithm again to find the best 2-dimensional solution.
- Proceeding in this way, find solutions with increasingly higher dimensionality. Stop, when the solution with dimensionality $n+1$ is a worse approximation than the solution with dimensionality n .

This approach is successful because the solution with dimensionality n is always an *approximate superset* of the solution with dimensionality $n-1$. By this we mean that $n-1$

of the objects in the solution with dimensionality n are always *very close* to the objects in the solution with dimensionality $n-1$. *Very close* in this case means that it is generally a matter of hill-climbing to move from the positions of the $n-1$ objects in the $n-1$ dimensional solution to the optimum positions of the “corresponding” $n-1$ objects in the n -dimensional problem. Thus the biggest challenge in moving from dimensionality $n-1$ to dimensionality n is to determine where to put the additional object.

4.3 Genetic Operators

We defined the following genetic operators. The first two are combination operators; the last two are mutation operations

1. **Cross-over.** Select two elements. Select objects from each to include in the third element. Recall that during each run of the genetic algorithm, the population is homogeneous in size: each element has the same number of objects. Furthermore, the objects are sorted by x-position. This makes cross-over a more meaningful operation since comparable objects are being substituted for each other.
2. **Weighted-average of elements.** This operator is similar to crossover. But instead of selecting objects randomly from either of the two parents, create a new object by taking the weighted average of the corresponding elements in the parents. The weights depend on how good an approximation the parents are. This turns out to be a powerful hill-climbing operation.
3. **Line-optimize an object.** Given an element, select at random both an object and a direction in the x-y plane. Using standard line-optimization techniques, move the object along the selected direction until one finds a position that maximizes the element’s overall value.

4. Brethorst's optimization technique.

Brethorst [Br87] has developed a powerful optimization technique for problems similar to this one. In many cases, it finds the optimum value on its own. Our experience was that in some cases it found only a local optimum. We therefore incorporated it as an operator in our genetic algorithm framework.

To use it, select an element from the population, which is used to seed the Brethorst algorithm. The result produced by the Brethorst algorithm is taken as the result of the operator.

The integration of known optimization techniques into a genetic algorithm framework poses a challenge. That challenge and our approach to its resolution is discussed in the following section.

4.4 Maintaining Population Diversity

A fundamental principle of genetic algorithms population management is: the better an individual, the better its chance of being retained in the population. A consequence of this principle is that over time, even if new population elements were selected from the search space at random, the average fitness of the population would increase.

A second fundamental (and countervailing) principle of genetic algorithms is the need to maintain diversity. Premature population convergence to a suboptimal solution is exactly what genetic algorithms are intended to avoid.

We have developed an approach to population management that attempts to satisfy both objectives. Our approach is based on the use of two techniques: continual injection of new, random elements into the population and tournament selection with varying competition levels.

Continual injection of new, random elements is just as it sounds. Instead of transforming an existing schedule, an entirely new schedule is

generated. This ensures that the population will never be completely isolated in one part of the search space. In addition, once a new random element is generated one of the two mutation operations are applied to it to allow it climb to a local maximum. This increases the probability that the element will be retained in the population long enough to participate in additional transformations.

Tournament selection is used in the transformation and replacement step. It is used to select both the element(s) to be transformed and the element to be discarded.

To select an element for transformation, a subset of the population, *the selection pool*, is chosen randomly and uniformly from the entire population. The best (or best two) element(s) of that pool are selected. To select an element to be discarded, we again choose a subset of the population; the worst element of the selection pool is selected for deletion.

Since elements are included in the selection pool with equal probability, the size of the selection pool is *inversely related* to the selectivity of the search. If the pool size were 1, one would be selecting (for transformation or deletion) an element uniformly from the population, i.e., with no regard for how well the element solved the problem. This would minimize convergence, but it would also minimize the likelihood that good features would be exploited.

On the other hand, were the pool to be the entire population, one would always select the best element(s) for transformation and the worst for deletion. This would maximize convergence, but it would virtually eliminate significant diversity.

Our strategy is to allow the size of the selection pool to vary based on the extent to which the population has converged. Convergence is measured by the difference between the best element of the population and the median element of the population. As they approach each

other, the size of the selection pool is decreased, thereby promoting population divergence. As they move apart, the size of the selection pool is increased, thus promoting population convergence.

This yo-yoing effect tends to ensure that the population will not converge to a single area of the search space.

5.0 Results

The genetic algorithm strategy was applied to real data from a staring visible CCD sensor attached to a 24-inch telescope on Table Mountain, California. Figure 2 shows the center 256X256 pixel scene of NGC 6819 measured September 19, 1992. We chose four stars as model PRFs and ran 1-4 source models on 32 different clumps. The clumps were chosen to include single and multiple sources with a variety of amplitudes at locations.

Three cases are discussed. In these examples, the bright star at $\sim(250, 320)$ was used as the PRF. The three clumps are displayed in Figure 3 as detailed contour plots.

- **Star clump 423 at $\sim(300, 190)$** is commonly thought to be a single source and is used for calibration photometry. Our algorithm agreed with this hypothesis with an extremely high confidence of 99%. The estimated amplitude was 1198 counts with an error bar of only 3 counts. The location was 9.704 ± 0.006 pixels east and 11.493 ± 0.006 pixels north. The small error bars result from the high signal to noise ratio of ~ 150 .
- **Star clump 416 at $\sim(200, 360)$** is also commonly taken to be a single source. Our technique, however, assigned virtually no probability to a one-source model compared to a two source model. The two source model was also preferred over the three source model by a factor of 100. The two source model put a source about 11

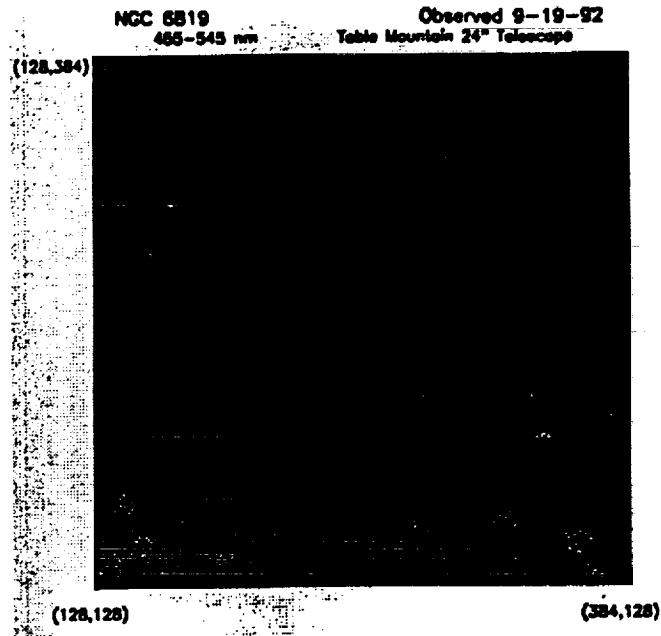
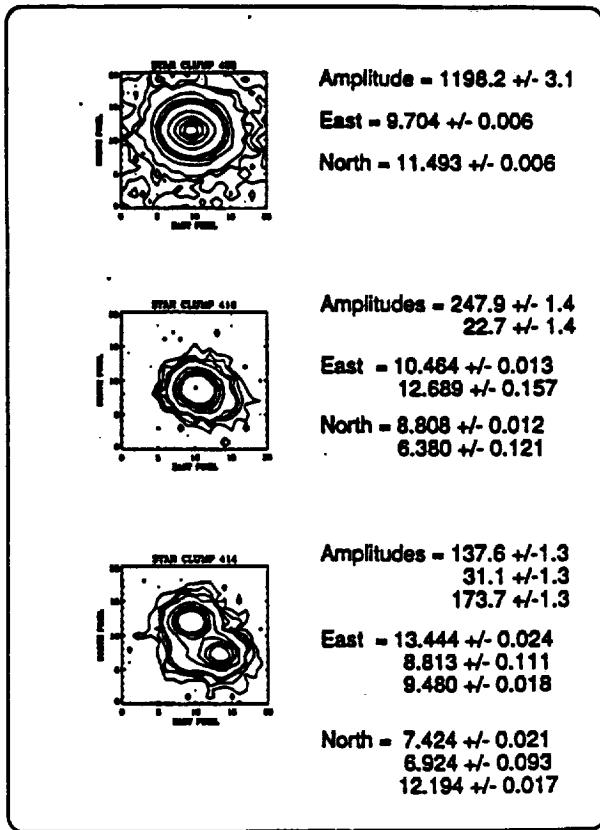


Figure 2. View from Table Mountain

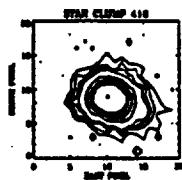
times dimmer than the other separated by 2.2 pixels to the east and 2.5 pixels to the south. The location error bars are greater than $\sim 1/6$ pixel for the dimmer source compared to 0.013 pixel for the brighter source. The amplitude error bars for the dimmer source were about 6% compared to 0.5% for the brighter source.

- **Star clump 414 at $\sim(340, 210)$** looked interesting because the bulge to the south and west of the doublet gives evidence for another source. The technique, in fact, strongly preferred a three source model with a dim source located at 8.8 pixels east and 6.9 pixels north. It was separated by 4.6 pixels east and 0.5 pixels south from one source and 0.6 pixels east and 5.2 pixels south from the other pulse.

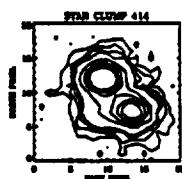
For all of the above clumps, we ran the code several times with different initial guesses. In all cases the genetic algorithm converged on the indicated solution in a reasonable amount of processing time. A standard hill-climbing



Amplitude = 1198.2 +/- 3.1
East = 9.704 +/- 0.006
North = 11.493 +/- 0.006



Amplitudes = 247.9 +/- 1.4
22.7 +/- 1.4
East = 10.484 +/- 0.013
12.689 +/- 0.157
North = 8.808 +/- 0.012
6.380 +/- 0.121



Amplitudes = 137.6 +/- 1.3
31.1 +/- 1.3
173.7 +/- 1.3
East = 13.444 +/- 0.024
8.813 +/- 0.111
9.480 +/- 0.018
North = 7.424 +/- 0.021
6.924 +/- 0.093
12.194 +/- 0.017

Acknowledgments

This work was performed at the Aerospace Corporation under sponsorship of its Aerospace Sponsored Research Program. We thank M. Campbell for many fruitful discussions.

References

- [Br87]. Bretthorst, L, *Bayesian Spectrum Analysis and Parameter Estimation*, Ph.D. Dissertation, Washington University, St. Louis, 1987.
- [Da87]. Davis, L and S Coombs, "Genetic algorithms and communication link speed design: theoretical considerations," *Proceedings of the Second International Conference on Genetic Algorithms*, pp 252-256, 1987.
- [Da91]. Davis, L (Ed), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.
- [Fi84]. Fitzpatrick, JM, JJ Grefenstette, and D Van Gucht, "Image registration by genetic search," *Proceedings of the IEEE Southeast Conference*, pp 460-464, 1984.
- [Go83]. Goldberg, DE, *Computer-Aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning*, Doctoral Dissertation, University of Michigan, 1983.
- [Gr85]. Grefenstette, JJ, R Gopal, B Rosamita, DV Gucht, "Genetic algorithms for the traveling salesman problem," *Proceedings of International Conference on Genetic Algorithms and their Application*, pp 160-165, Lawrence Erlbaum & Assoc., NJ 1985.
- [Ho75]. Holland, JH, *Adaption in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, Michigan, 1975.
- [Li94]. Lillo, W and NW Schulenburg, A *Bayesian Closely Spaced Object Resolution Technique*, Aerospace Technical Report ATR-94(8025)-1, 1984.
- [Sc93]. Schulenburg, NW, and JA Hackwell, *Bayesian Approach to Image Recovery of*

technique gave different solutions for different initial guesses.

6.0 Conclusion

This work has shown that a hybrid genetic algorithm can be developed to solve a difficult optimization problem arising from image processing. Our experience has convinced us that neither traditional optimization techniques nor traditional genetic algorithm techniques would have allowed us to solve this problem. As our genetic operators show, a hybrid genetic algorithm approach allows one to incorporate known optimization techniques into a genetic algorithm framework. This strategy demonstrates that one need not sacrifice known, powerful techniques when one employs genetic algorithms.

Closely Spaced Objects, SPIE Conference, Orlando, Florida, April 12-14, 1993.

[Wh89]. Whitley, D and T Hanson, "Optimizing neural networks using faster, more accurate genetic search," *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 391-396, Morgan Kaufmann, San Mateo, Ca. 1989.

Improved Reduced-Resolution Satellite Imagery

James Ellison and Jaime Milstein
The Aerospace Corporation
El Segundo, California 90245

5/3-61
47263
P- 16

Abstract

The resolution of satellite imagery is often traded-off to satisfy transmission time and bandwidth, memory, and display limitations. Although there are many ways to achieve the same reduction in resolution, algorithms vary in their ability to preserve the visual quality of the original imagery. These issues are investigated in the context of the Landsat browse system, which permits the user to preview a reduced resolution version of a Landsat image. Wavelets-based techniques for resolution reduction are proposed as alternatives to subsampling used in the current system. Experts judged imagery generated by the wavelets-based methods visually superior, confirming initial quantitative results. In particular, compared to subsampling, the wavelets-based techniques were much less likely to obscure roads, transmission lines, and other linear features present in the original image, introduce artifacts and noise, and otherwise reduce the usefulness of the image. The wavelets-based techniques afford multiple levels of resolution reduction and computational speed. This study is applicable to a wide range of reduced resolution applications in satellite imaging systems, including low resolution display, spaceborne browse, emergency image transmission, and real-time video downlinking.

1 Background

Satellite imaging systems like Landsat, collect and downlink large quantities of data. Associated ground systems may further process and store this data, as well as provide for its dissemination. Limitations on computer storage, transmission bandwidth, transmission time, and digital display resolution may restrict the amount of data used to represent an image. These issues affect image processing and storage on-board the satellite, preparation of the image for transmission, downlinking of image data, and reconstruction, storage and dissemination of the image to the end user. Such problems may be addressed by data compression techniques, by reducing image coverage, by reducing the number of gray levels (or colors), or by reducing

resolution. Some resolution-reducing techniques (for instance, edge-avoiding convolution) are scene-dependent. This paper considers only general resolution reduction algorithms. In particular, wavelets, a recently developed mathematical transform, is utilized as a resolution-reducing device and compared with some conventional algorithms for resolution reduction.

Section 2 discusses an example of a typical problem requiring resolution reduction. Some common methods for handling the problem are discussed, and the idea of wavelets is introduced. A quantitative measure is used for crude quality comparisons. Potential applications of a good solution to the resolution reduction problem are also suggested. In Section 3, resolution reduction algorithms are applied

to Landsat imagery, and numerical comparisons are given. Based on visual examination, experts^{*} concluded that wavelets preserves image quality better than other methods tested. In Section 4, aerial images are used to illustrate the visual quality resulting from alternative methods. Conclusions are summarized and applications are suggested in Section 5.

2 Reducing Image Resolution

In this section we briefly discuss resolution-reducing algorithms based on subsampling, convolution and wavelets. We conclude by noting the applicability of a good resolution-reducing algorithm to other practical problems.

2.1 An Example of a Problem in Resolution Reduction

Suppose we wish to display a full-size M -by- N pixel image on a P -by- Q pixel screen, $P \ll M, Q \ll N$. This problem arises, for example, when the full 5984-by-6200 pixel scene presented by the Landsat Thematic Mapper (TM) is to be displayed on a conventional personal computer monitor, which may permit up to 512 rows and 650 columns. Under these circumstances, it is impossible to display the full scene on the pixel-limited display without sacrificing *resolution*, the minimal distance at which small adjacent objects can be distinguished [Rosenfeld and Kak, 1982, p. 215]. In this example, the original 5984-by-6200 pixel scene has 16 times the resolution of a 374-by-388 pixel rendition of it.

The visual degradation of a reduced-resolution image depends on the resolution reduction technique. Our goal is to reduce resolution in such a way that the eye's perception of the displayed scene is as close as possible to that of the full resolution scene. This is what we mean by the "display problem".

The resolution-reducing algorithms discussed below have a common property which enables us to compare the reduced resolution imagery they produce: each algorithm can be represented as a series of applications of a 2-to-1 resolution-reducing technique, whether subsampling-by-2, wavelets, or some other methodology. If 2-to-1 resolution reduction is applied k times, then the algorithm produces a 2^k -to-1 resolution-reduced image, directly comparable to the image produced by applying any other 2^k -to-1 resolution-reducing algorithm. For example, since subsampling-by-16 amounts to 4 iterations of subsampling-by-2, it is reasonable to make quality comparisons between the results of subsampling-by-16 and that of applying 4 iterations of wavelets to the same original image: the resulting images have the same resolution and differ only in the algorithm applied.

We now focus on subsampling and wavelets. Each provides a practical, computationally efficient solution, independent of scene, subject matter, and prior degradation. Yet, subsampling and wavelets represent opposite extremes of mathematical soundness and visual appearance.

* The imagery discussed in this paper was presented to image exploitation professionals and other scientists of the EROS Data Center of the U. S. Geological Survey, together with a wide variety of scientists from the Landsat user community. "Expert conclusion" refers to the unanimous opinion of this population.

2.2 Subsampling

The most straightforward way to reduce the size of the array without losing coverage is by *subsampling*, that is, assembling an image from a regularly spaced subset of pixels in the original array. *Subsampling-by-n* involves the selection of pixels from every n th column along every n th row. Thus, subsampling-by- n results in an n^2 -to-1 reduction in the number of pixels and an n -to-1 reduction in resolution.

Subsampling is widely employed as an efficient solution to the display problem. For example, as noted in Section 3, subsampling-by-16 is currently employed in preparing the Landsat browse product for the user community from the original TM scene. In principle, subsampling requires no computation and is therefore optimal in computational efficiency.

However, visual defects are introduced by subsampling-by- n . As n becomes a significant fraction of the width (in pixels) of any feature, these defects worsen. The following defects are typical of imagery produced by subsampling.

- Edges of solid bodies assume a staircase appearance.
- Even when a feature covers most of an n -by- n square, this information is lost if the sampled pixel happens not to fall within the feature.
- Separate, distinct features can merge.
- Linear features, i.e., long narrow features like roads, communication lines, and rivers, can disappear altogether.

- Small features can vanish.

- When the retained pixel is unrelated to its surroundings, this pixel shows up in the reduced resolution image as apparent noise.
- Artifacts can be introduced by random noise. As noise increases, larger artifacts become more common.

As resolution is reduced, some loss of image quality is unavoidable. However, much of the loss of visual quality just described is peculiar to the subsampling process itself. The obvious problem with subsampling is that the retained pixels provide no information about the discarded pixels. Generating the same amount of data, more effective resolution-reducing methods capture more representative visual data from the full resolution image than does subsampling. Instead of picking one pixel out of a fixed position in the n -by- n square, they define a value of the new pixel that better represents the pixel values in the n -by- n square it is replacing.

2.3 Convolution

Convolution, or *spatial filtering*, creates a new image by replacing each pixel value with a weighted average of its surrounding pixel values. As a resolution-reducing technique, convolution may be regarded as a generalization of subsampling, in which a convolution is performed at each subsampled point. The corresponding pixel in the new image is given the value of the convolution. When that convolution is the unit impulse function (1 surrounded with 0's), this process reduces to subsampling-by- n .

Convolutions have been tailored to widely varying purposes, including edge enhancement, smoothing, noise reduction, etc. Convolution has also been combined with other algorithms for selective application to scenes or parts of scenes. Depending on coefficients of the convolution, the pixels in the reduced scene may retain useful information from those discarded from the original scene. For this reason, the resulting image may be less subject to many of the defects characteristic of subsampling.

Computation required for any specific convolution is proportional to the number MN of pixels in the original scene: each of the PQ pixels in the n -to-1 reduced resolution image represents up to n^2 multiplications and additions, and $(n^2)PQ = MN$. Convolution offers a fast method for resolution reduction, though not as fast as subsampling. However, in our experience, for a specific convolution, apparent degradation typically varies greatly, depending on the nature of the scene, its texture, etc.

2.4 Wavelets

Wavelets may be regarded as a special kind of convolution. Wavelets developed rapidly from 1983 onward. There is now a large and rapidly growing literature on the subject [Meyer, 1986; Mallat, 1989; Chui, 1991; Press, 1991]. The present work uses coefficients defined by Daubechies [Daubechies, 1988]. Our purpose here is to discuss wavelets only to the extent necessary to provide a context for the present application.

As commonly employed, the term "wavelets" refers to a data compression technique with many elegant properties,

both theoretical and practical. When applied to an image represented by a $2P$ -by- $2Q$ array, wavelets generates four P -by- Q arrays. One array, called the *smooth image*, is a reduced resolution version of the original image. The values in the present study were Daubechies's D_4 coefficients (or weights): $\frac{1}{4}(1+\sqrt{3})$, $\frac{1}{4}(3+\sqrt{3})$, $\frac{1}{4}(3 - \sqrt{3})$, $\frac{1}{4}(1 - \sqrt{3})$.

The computation time required for wavelets is, like convolution, proportional to the number of pixels in the original image. Used for resolution reduction, the number of pixels dealt with in each iteration of wavelets is $\frac{1}{4}$ that of the previous iteration. Thus, k iterations of wavelets, applied to an M -by- N pixel image, has a computation time proportional to $MN[1 + \frac{1}{4} + \dots + (\frac{1}{4})^{k-1}]$. Since $[1 + \frac{1}{4} + \dots + (\frac{1}{4})^{k-1}] < 1\frac{1}{2}$, for all positive k , the computation time for wavelets resolution reduction remains proportional to the number of pixels in the original image, independent of the size of the final reduced-resolution image. (In practice, clever implementation can significantly reduce the amount of computation.)

Only the smooth images are needed for the purpose of resolution reduction. Thus, k iterations of wavelets resolution reduction generate an image of the same 2^k -to-1 resolution reduction as k iterations of subsampling-by-2 (i.e., subsampling-by- 2^k). The results of these algorithms are compared in Sections 3 and 4.

In a certain well-defined sense, for a given resolution reduction 2^k -to-1, k iterations of wavelets better preserve image quality and are not prone to pronounced artifacts such as those associated with subsampling.

2.5 Some Reduced Resolution Problems

Any technique that leads to better quality reduced-resolution imagery has many potential applications in Landsat and other satellite imaging systems. A few such applications are noted below.

The Landsat display problem

A full resolution image from the current Landsat Thematic Mapper typically requires a 5375-by-6468 pixel array. Any attempt to display such an image on a computer monitor, even one capable of a 1012-by-1012 display, requires a solution to the display problem. Moreover, a flexible solution would permit the individual user to tailor the final resolution to his or her display capability.

Landsat browse

The full resolution Landsat TM image consists of approximately 280 megabytes of data, about 40 megabytes per spectral band. The Landsat browse product is reconstituted for the user on location from data transmitted over phone lines or the Internet. Currently, three bands of data are reduced from 40 megabytes to 156 kilobytes per band using subsampling-by-16, to produce a false color, reduced-resolution version of the original image of about 335-by-404 pixels. This process avoids most of the data storage and transmission that would otherwise be required. Based on a "quick look" at the resulting image, the user can then request (and pay for) full-detail imagery of interest. A superior solution is one that gives better quality imagery of the same resolution than currently available. It would also be useful for the user to be able to select from a range of resolution-reductions. This would add to the current

full-resolution and 1/16th resolution alternatives a range of cost and bandwidth-intensive choices.

Downlink browse

This application postulates a high resolution satellite sensor with a downlink bandwidth constraint. The principle of operation is similar to that of the Landsat browse: the satellite downlinks a reduced resolution image for approval before transmitting (or even collecting) the full resolution image. This way, depending on the image and resolution desired, downlink bandwidth can be used or conserved.

Emergency spaceborne image communication

The downlink of a spaceborne remote sensing system could be jammed or otherwise dysfunctional. In this case, the satellite could be instructed to transmit a reduced resolution image to a communication satellite network for retransmission and downlinking.

Animation or real-time video downlinking

This scenario envisions the adventure movie scenario of an interactive capability enabling an imaging satellite to zoom in on a selected target area. Frequent images (animation) or real-time video would then be downlinked. Among the challenges in designing such a system is that of limited downlink bandwidth. However, the human eye is more forgiving of reduced resolution when viewing animation and video than when examining an individual image. This facilitates trade-offs of resolution reduction in favor of frame frequency. Suppose, for example, that the system has a 0.1 meter earth surface resolution and can downlink 24 megabytes of imaging data per second, with the ability to take and process up to 24 frames

per second. Such a system might be able to transmit a full-resolution, single-band Landsat-quality scene in 1.6 seconds. This system could instead be instructed to transmit 24 frames of a 100-by-100 meter square of the earth surface per second, at full one meter resolution. A good (and fast) 16-to-1 resolution-reducing algorithm might provide interactive real time video coverage of a 1.6 kilometer square.

3 The Landsat Browse Study^{**}

3.1 The Current Browse Product

The current browse product provides users with an economical reduced resolution preview of Landsat imagery. The browse product is reconstituted on location from data transmitted over phone lines or the Internet. From this preview, the user decides whether or not to request full-resolution imagery.

Currently, subsampling-by-16 is applied to 3 bands of full resolution Landsat imagery in order to provide a single RGB reduced-resolution browse product of about 335-by-404 pixels. The user thus views 0.4% of the pixels from each of three bands of the original full resolution image. The subsampling deficiencies discussed in Section 2.2 are readily apparent in practice, as seen in imagery found in the next section.

The object of the study was to develop and investigate resolution-reducing algorithms that produce superior quality browse imagery over the full range of geographic scenes. In particular, such deficiencies in the current browse product as the potential loss of linear features should be overcome. The investigators established three ground-rules as fundamental to the study:

- Every candidate algorithm must produce browse images of the same resolution as those generated by the current system.
- In order to make the browse product available to the user in near-real time, the computer processing required to generate the browse image must not add more than 3 minutes to the total service delay.
- The browse product must be effective with the full range of geographic imagery.

In the course of the study, several algorithms were investigated: subsampling, wavelets, 3-by-3 convolution, and various hybrid algorithms. These algorithms varied in the quality of the resulting browse product and in processing time.

In the final study phase, experts visually compared the various browse images and products both to the full resolution image and to one another. In the earlier study phases, resolution-reduced images were compared in terms of an objective measure we now describe.

^{**} This work was conducted at The Aerospace Corporation in 1992-1993 with funds provided by NASA Goddard Space Flight Center. Dr. M. Jankins, now at Disney Feature Animation, assisted the project at every stage with his thorough understanding of wavelets. Dr. Jankins also provided extensive software development and programming support, both for prototyping and experimentation. This study was conceived when Milstein perceived the wavelets smooth image as a possible solution to the Landsat browse problem.

3.2 A Measure of Image Degradation

This study used a quantitative measure we refer to as the *sequential correlation coefficient (SCC)*, defined as the average correlation between the intensity at a pixel and that of its immediate neighbor on the right. (This measure is not to be confused with more sophisticated imagery measures involving two dimensional statistical correlation.) The SCC can be used as a crude measure of image degradation. In principle, the SCC can assume any value between -1 and 1, the more positive the value, the less the average change. For example, the SCC of an image is 1 if all pixels in the same row have the same intensity. An SCC of 0 suggests a completely random "snow-like" image (e.g., the TV screen when a channel is not broadcasting). For practical purposes, the SCC of recognizable imagery is generally well above .60.

For insight into the significance of the sequential correlation coefficient, compare almost any scene or picture of interest to "snow". A "real" scene tends to be a patchwork of regions and well-defined objects or features. Two adjacent pixels are more likely than not to fall within the same feature or region, have similar intensity, coloring, etc. In the "snow" scene, however, even adjacent pixels are likely to be dissimilar. For any "real" scene the greater the distance between the pixels, the more likely they are to fall into different regions or features, having unrelated, widely varying colors (intensities in various bands). Thus, for any real scene, as the resolution is reduced, the SCC can be expected to decrease. This is clearly true of subsampling-by- n , as n increases.

The SCC is not a completely reliable measure of image quality as interpreted by the eye. For example, "turning down the contrast" of an otherwise good quality image can reduce the eye's perception of quality while increasing the SCC. As a practical matter, a one or two-percent difference between SCCs is unpredictive of comparative visual quality.

In the early phases of the study, the SCC proved a useful heuristic for comparing image degradations caused by alternative reduction algorithms. Final conclusions were based on the judgment of expert viewers representing the user community and were consistent with the SCC-based findings.

3.3 The Three-Phase Browse Study

Phase 1 of the study was an assessment of a wide variety of candidate algorithms and an initial proof-of-concept of iterated wavelets as a resolution-reducing methodology. Phase 1 used *Landsat P data* - Landsat full resolution imagery after radiometric and geometric correction. Phase 2 investigated two additional algorithms, checked processing speeds, and extended the investigation to *Landsat raw data* (i.e., full resolution images not radiometrically or geometrically corrected). Phase 3 investigated two additional algorithms, each computationally faster than wavelets and more effective than subsampling. Table I surveys the algorithms tested in the course of the study. In addition to a wide variety of Thematic Mapper images, Phase 3 included digitized aerial imagery with resolutions higher than that of the current Landsat. Examples of these reduced resolution images are found in Section 4.

TABLE I
Reduced Resolution Algorithms
Investigated in Each Study Phase

Algorithm	Phase 1	Phase 2	Phase 3
Subsampling			
Subsampling-by-2	x		
Subsampling-by-4	x		
Subsampling-by-8	x	x	
Subsampling-by-16*	x	x	x
Subsampling-by-32	x	x	
Wavelets			
One iteration	x		
Two iterations	x		
Three iterations	x	x	
Four iterations**	x	x	x
Five iterations	x	x	
3 x 3 Convolution			
	x		
Hybrid			
SS-by-2, 3 wavelets iterations***	x	x	
SS-by-4, 2 wavelets iterations***		x	
SS-by-8, 1 wavelets iteration	x		

*The current algorithm for browse

**The candidate wavelets algorithm for browse

***The major candidate hybrid algorithms for browse

when iterated wavelets is compared to iterated subsampling-by-2. Data is shown for only one spectral band (Band 3) because band-to-band variation in the SCC was negligible in every case. With each iteration of subsampling-by-2, the SCC drops about 0.08 until, with the fifth iteration, the SCC falls below 0.60, the "threshold of intelligibility". By comparison, a single application of wavelets induces a loss of about 0.04. The next four applications of wavelets together result in an additional loss of about 0.03. (The slight increase in the SCC for lower resolution wavelets, though negligible, is an artifact of the crudeness of the SCC as a measure of image quality.) Consequently, after 5 iterations of wavelets the SCC is approximately that of one iteration of subsampling-by-2, while the SCC of the image resulting from subsampling-by-32 (5 iterations of subsampling-by-2) suggests a severely degraded image.

Landsat Browse Study - Phase 1

The Phase 1 study used a 5965-by-6967 pixel scene*** that included an urban setting having many linear features. RGB false color images were generated using Bands 5, 4, and 3 of the seven spectral bands obtained from the Landsat Thematic Mapper. Ten RGB images were generated, five by iterated subsampling-by-2 and five by iterated wavelets. The SCC was evaluated for each color (band) of each image.

As expected, the SCC tended to decrease with each application of subsampling-by-2 and with each application of wavelets to the Urban P-data scene. Figure 1 illustrates dramatically different behavior of the SCC

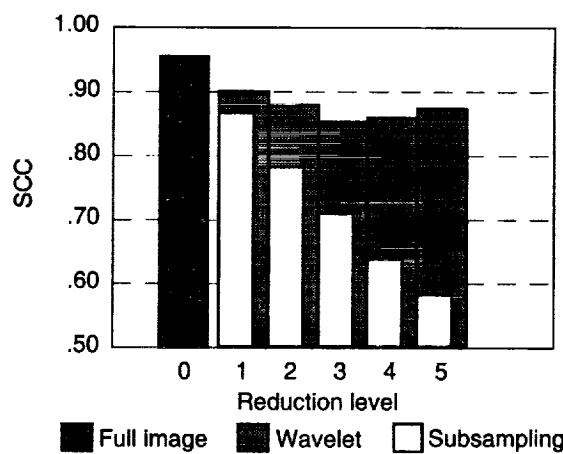


Figure 1. Comparison of Sequential Correlation Coefficients For Subsampling Band 3

The Phase 1 results showed that the wavelets approach is a good alternative to the present subsampling technique. Wavelets-generated imagery retained more

*** All Landsat P data and raw data used in this study were supplied by Stuart Doescher of the U. S. Geological Survey (USGS) EROS Data Center in Sioux Falls, SD.

features at reduced resolution and had fewer artifacts: in particular, linear features were never obliterated.

Landsat Browse Study - Phase 2

Phase 2 of the study used Landsat raw data to examine the robustness of wavelets in conserving image quality. This phase also addressed computation time issues. A conventional 3-by-3 convolution [Pratt, 1991, p. 303] was tested as a foundation for a browse capability (see Section 2.3). Milstein's *hybrid-1* technique was also investigated. This method, consisting of subsampling-2 followed by iterated wavelets, was expected to reduce processing time by 75%, compared to iterated wavelets alone.

The data used for this study consisted of Band 5, 4, and 3 raw data for two 5984-by-6400 scenes: a forested mountain scene and a scene consisting of clouds, water, and vegetation. As in Phase 1, five levels of reduction were applied to each scene, using each of the four algorithms. The resulting SCC values are shown in Figure 2 (the forested mountain scene) and Figure 3 (the clouds, water, and vegetation scene).

Compared to Phase 1 results, the degradation represented by the decline in the SCC for the two raw images is slightly greater for wavelets and significantly greater for subsampling, and there is noticeable variation from band to band. This is seen in Figures 4 and 5, which compare SCCs of the 16-to-1 reduced resolution images generated by the four algorithms. Otherwise, SCC findings for subsampling and wavelets do not differ very much from those of the Phase 1: the rapid degradation that occurs for subsampling greatly exceeds that of wavelets.

As suggested by Figures 2 through 5, the performance of the 3-by-3 convolution as a resolution-reducing technique was only marginally better than subsampling. However, the SCCs for hybrid-1 resolution reduction were nearly identical to their pure wavelets counterparts. This unexpected finding suggested hybrid-1 as a viable, high speed alternative to wavelets.

Phase 2 analysis also addressed the question of the relative sensitivity of the browse image to the uncorrected distortions in the raw image under the various algorithms. It was found that neither the wavelets algorithms nor the hybrid algorithms propagated the geometric or radiometric errors for any level of resolution. Both wavelets and hybrid methods proved robust, in particular, when applied to raw image data or to uncorrelated data. This finding dispelled concern for possible error propagation.

These algorithms were implemented by approximately 160 lines of C code. The runs on a Sun SPARC 10 Workstation showed that the run-time performance of all the algorithms meets Landsat 3-minute time constraint. For 16-to-1 resolution reduction, subsampling was by far the fastest algorithm ($\frac{1}{2}$ second for non-computational processing), followed by convolution and hybrid-1 (30 seconds), and wavelets (180 seconds).

Landsat Browse Study - Phase 3

Phase 3 investigated two additional algorithms, each computationally faster than wavelets and more effective than subsampling. Phase 3 used a wide variety of full resolution Landsat imagery, in addition to still higher resolution aerial imagery. The

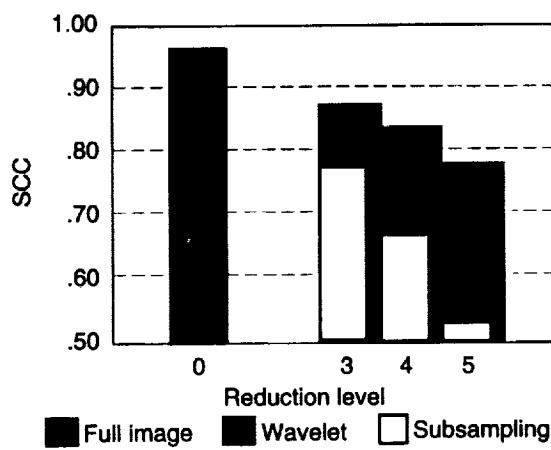


Figure 2. Comparison of SCCs for Subsampling and Wavelets Generated Imagery Band 3, Forested Mountain Scene, Raw Data

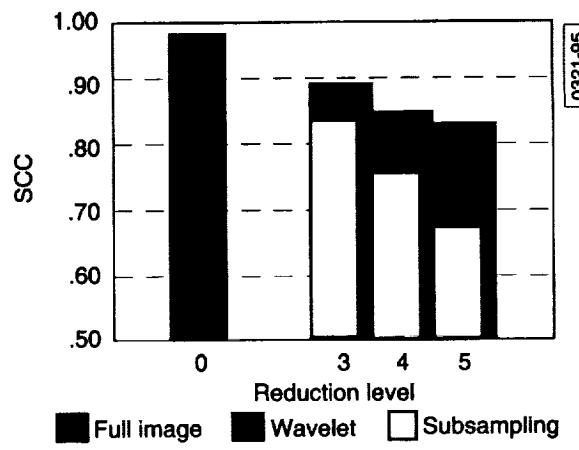


Figure 3. Comparison of SCCs for Subsampling and Wavelets Generated Imagery Band 3, Clouds, Water & Vegetation Scene, Raw Data

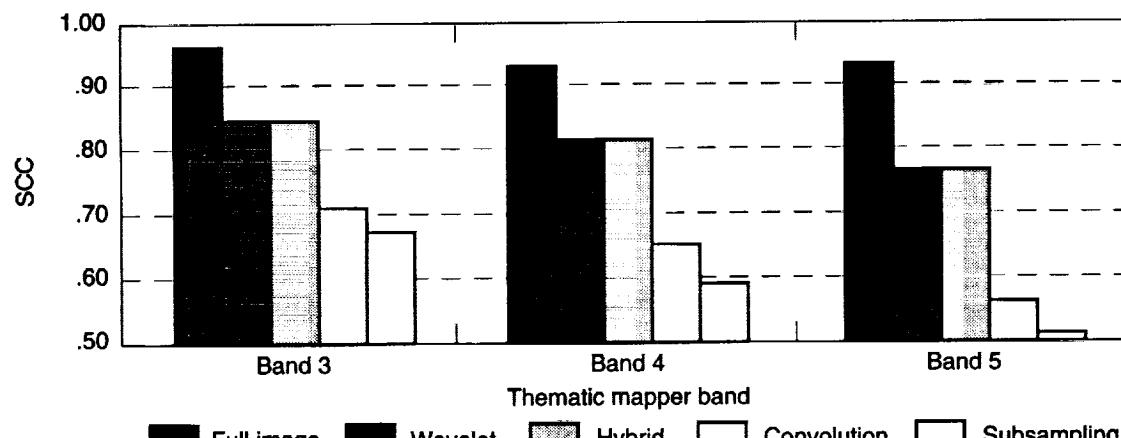


Figure 4. Comparison of SCCs for Four 16-to-1 Resolution Reduction Algorithms: Band 3, Forested Mountain Scene, Raw Data

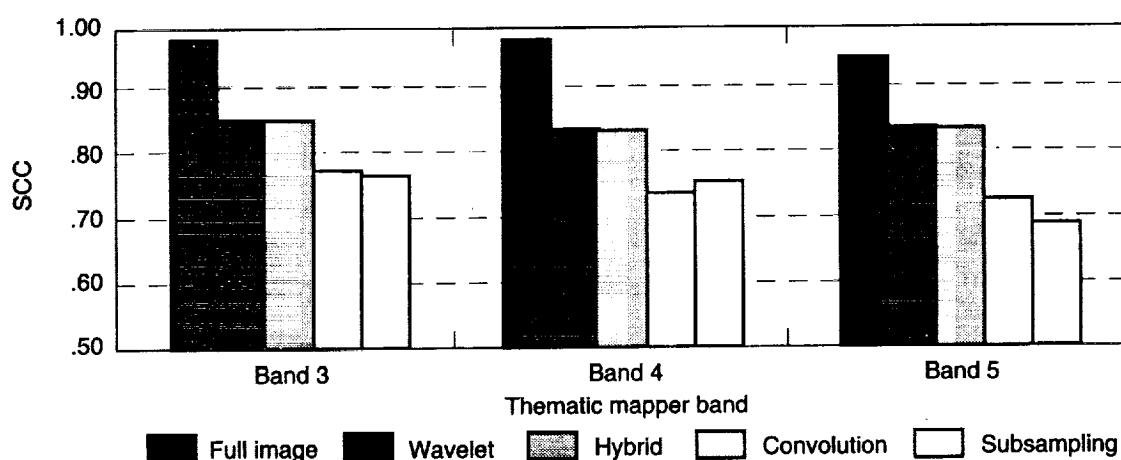


Figure 5. Comparison of SCCs for Four 16-to-1 Resolution Reduction Algorithms: Band 3, Clouds, Water & Vegetation Scene, Raw Data

aerial imagery is shown in reduced resolution and discussed in Section 4.

The major issues treated in Phase 3 were the investigation of two more hybrid algorithms and the comparison and evaluation by experts from the scientific community of 16-to-1 reduced imagery generated by alternative algorithms: subsampling-by-16, wavelets, hybrid-1, *hybrid-2* (subsampling-by-4, followed by two iterations of wavelets), and *hybrid-3* (subsampling-by-8, followed by one iteration of wavelets). Compared to iterated wavelets, *hybrid-2* and *hybrid-3* reduce the number of computations by factors of 16 and 64, respectively.

Experts found that 16-to-1 reduced resolution imagery produced by wavelets, hybrid-1, and hybrid-2 were virtually indistinguishable from one another, though slightly superior to *hybrid-3* imagery. All were found far superior to imagery produced by subsampling-by-16. Experts considered imagery produced by wavelets and the three hybrid techniques useful for various purposes, but agreed that imagery produced by subsampling-by-16 had little value except for cloud determination.

This three-phase study established that a Landsat browse product based on either wavelets or a hybrid methodology offers a significantly better quality browse product within the Landsat processing time requirements than the current subsampling-based system. The new techniques produce more trustworthy imagery which can be stored and transmitted efficiently. Roads, communication lines, power lines, rivers, and other linear features are much better preserved by wavelets and the hybrid algorithms, and there are seldom artifacts.

Furthermore, these new methodologies provide greater flexibility, with the potential to meet future image reduction requirements arising from higher resolution imagery created by new sensor technologies.

4 Examination of Gray Scale Images

We now discuss a few reduced resolution aerial images used in the final phase of browse study. Figures 6, 7 and 8 show 16-to-1 reduced resolution versions of an aerial scene. This scene of an Air Force base, includes many roads and paths, a small runway, hills, and so forth.

Figure 6 shows the image after applying four iterations of wavelets to the full resolution image. All roads are clearly discernible, although there is some fade in-and-out or striation along the principal roads. Detailed hillside contour and erosion patterns are visible. It seems possible to make out much of the detail within the base itself. The SCC of this image is 0.90, compared to the full resolution image SCC of 0.98.

Now we examine Figure 7, the same resolution image, produced via subsampling-by-16. The road along the left edge of the military base has become a series of short, disjoint vertical segments, not much different in shape or intensity from horizontal segments just to their right. The same problem exists to varying degrees along most roads. Although the original image was virtually free of noise, the subsampled version has taken on a very noisy appearance, especially within the base area, where small features could assume the greatest importance to the user. This same "pseudo-noise" has washed out much of the



Figure 6. Reduced Resolution Airbase Image After Four Iterations of Wavelets

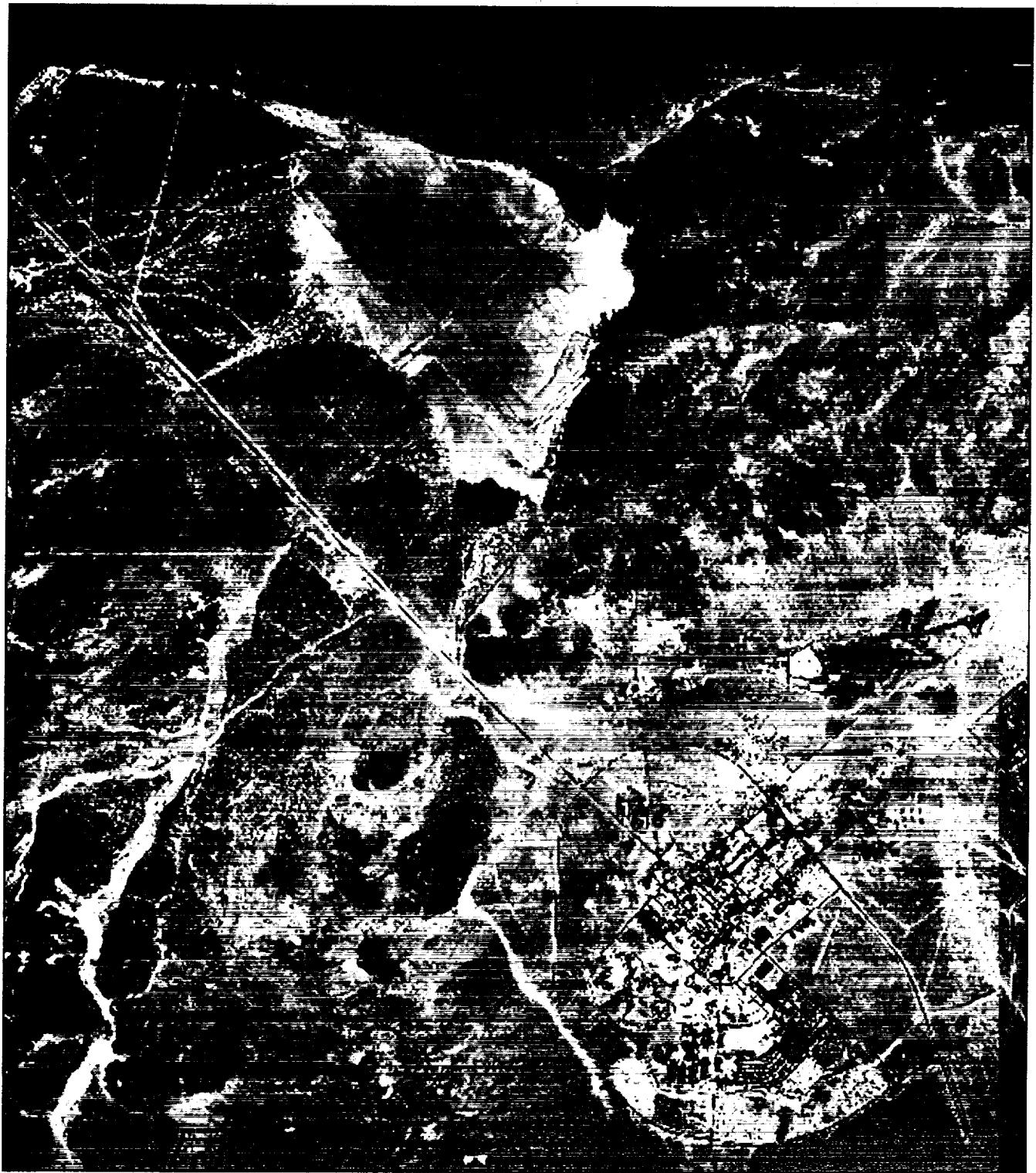


Figure 7. Reduced Resolution Airbase Image After Subsampling-by-16

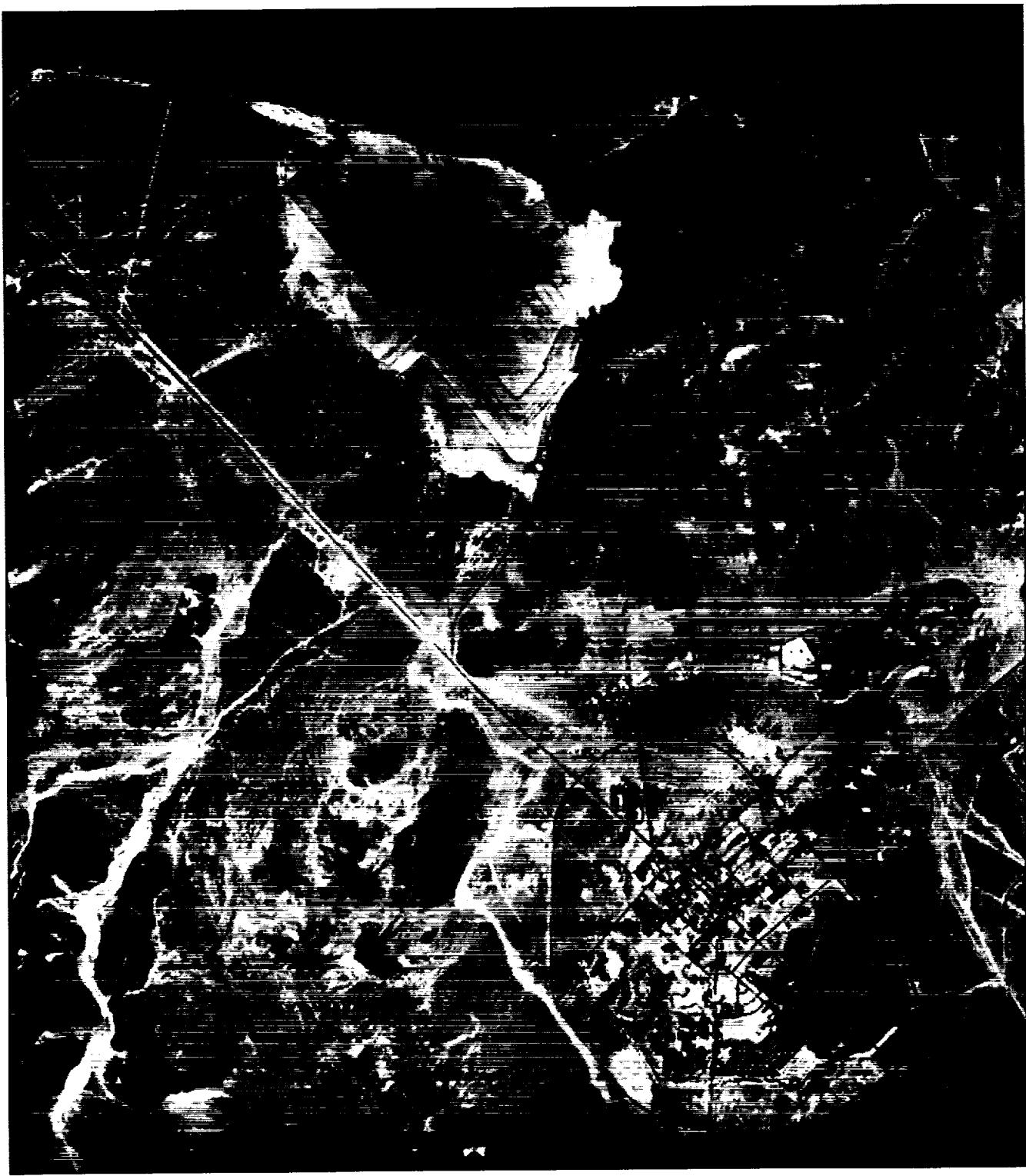


Figure 8. Hybrid-3 Reduced Resolution Airbase Image

topographical hillside detail found in Figure 6. If there had been significant random noise in the original image, the subsampled image would have been much more seriously degraded. The SCC of this image is 0.77.

In Figure 6, on the periphery of the base, about 3 inches from the left and 2 inches from the bottom of the image, is a small array of white objects. Even if we cannot identify this feature, we can use it as an aid in comparing the images. In Figure 7, we see that this feature is distorted beyond recognition (not surprising in view of the discussion in Section 2.2).

Figure 8 shows the effect of the hybrid-3 algorithm: subsampling-by-8, followed by wavelets. Under close scrutiny, we see slight but definite degradation, compared with the iterated wavelets image (Figure 6). For example, the small array is still visible, but the viewer is less certain as to its boundary. Yet, overall image quality seems much closer to pure wavelets than to pure subsampling. In fact, the SCC of this image is 0.89, compared to 0.90 SCC value for the wavelets image. In view of the visual quality of the hybrid-3 image, and the processing speed of the hybrid-3 algorithm (64 times that of wavelets) this algorithm could be an attractive alternative to iterated wavelets when computational speed is important.

The hybrid-1 and hybrid-2 images are not reproduced here. The hybrid-1 image appears visually indistinguishable from the pure wavelets image. The hybrid-2 image is distinguishable from the pure wavelets image but only in the finest of visible detail. There is no significant difference in the SCC values for wavelets, hybrid-1, and hybrid-2.

The quality of the hybrid-1 and hybrid-2 products, together with their processing speed-ups (respectively 4-to-1 and 16-to-1) compared to that of iterated wavelets, again make them serious alternatives to iterated wavelets in many applications. As a group, iterated wavelets, hybrid-1, hybrid-2, hybrid-3 constitute a prepackaged trade-off set of algorithms, which could give the user the luxury of choosing his or her own speed-quality trade-off.

5 Summary and Applications

The resolution of an image is the distance required between small objects in order to distinguish them from one another. In satellite imaging systems it is often desirable to generate reduced resolution versions of satellite imagery. Some deterioration in the visual quality of the imagery inevitably results from this process. However, some resolution-reducing algorithms are more effective than others in preserving the visual quality of the original image. We noted that a resolution reducing algorithm that does a good job in retaining visual quality has many potential applications to satellite imaging systems.

We recounted a study in which a variety of resolution-reducing algorithms were investigated in an effort to provide a superior browse product for Landsat imagery. Using a crude quantitative measure, we compared the current technique, subsampling-by-16, to a resolution-reducing technique based on a conventional convolution, an iterated wavelets-based algorithm, and several hybrid algorithms involving subsampling followed by iterated wavelets. Comparing images of the same resolution, those produced by iterated wavelets had quantitative measures

superior to those resulting from convolution and still more so from subsampling. The hybrid algorithms ranged from faster, with imagery visually indistinguishable from that of iterated wavelets, to much faster, with imagery of slightly lower quality than that produced by iterated wavelets. Imagery produced by pure subsampling was distinctly inferior compared to that of wavelets or any of the hybrid algorithms. Visual inspection by experts confirmed the findings suggested by the quantitative measure. Each of these algorithms can support resolution reductions of 2^k -to-1, $k \geq j$ ($j = 0$ for iterated wavelets, i for hybrid- i , $i = 1, 2$ or 3). The new algorithms were validated using the full variety of Landsat TM data, both P data and raw data, as well as higher resolution aerial imagery. All ran fast enough to satisfy browse requirements.

The wavelets-hybrid set of algorithms provide a speed-selectable set of 2^k -to-1 resolution reduction algorithms ($k = 0, 1, \dots$) applicable to a variety of imaging satellite system problems, including the Landsat display problem, the downlink browse problem, emergency spaceborne image communication, and real-time video downlinking, in addition to the Landsat browse problem.

Dedication

The authors dedicate this paper in honor of U. C. Berkeley Professor Hans J. Bremmermann, whose work on Haar functions pioneered early developments in the theory and application of wavelets.

References

- Daubechies, I. (1988). Othonormal Bases of Compactly Supported Wavelets. *Comm. Pure Appl. Math.*, Vol. 41, pp 909-996.
- Mallat, S. G. (1989). Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(\mathbb{R})$. *Trans. Amer. Math. Soc.*, Vol. 315, pp 69-87.
- Meyer, Y. (1986). Ondelettes et Fonctions Splines. Presented at the *Seminare Equations aux Dérivées Partielles*, Ecole Polytechnique, Paris.
- Muller, J-P, ed. (1988). *Digital Image Processing in Remote Sensing*. Taylor and Francis, Inc., Philadelphia.
- Pratt, W. K. (1991). *Digital Image Processing, Second Edition*, p 301. John Wiley and Sons, NY.
- Rosenfeld, A. and Kak, A. C. (1982). *Digital Picture Processing, Vol. 1, Second Edition*. Academic Press, Inc., NY.
- Strang, G. (1989). Wavelets and Dilation Equations: a Brief Introduction. *SIAM Review*, Vol. 31, No. 4, pp 614-627.

1995120968

N95-27389

**Reconstruction of an Infrared Band of
Meteorological Satellite Imagery
with Abductive Networks¹**

5/4-47

47264

P. 13

Harvey A. Singer, John E. Cockayne and Peter L. Versteegen
Science Applications International Corporation (SAIC)
1710 Goodridge Drive
McLean, Virginia 22102

Abstract.

As the current fleet of meteorological satellites age, the accuracy of the imagery sensed on a spectral channel of the image scanning system is continually and progressively degraded by noise. In time, that data may even become unusable. We describe a novel approach to the reconstruction of the noisy satellite imagery according to empirical functional relationships that tie the spectral channels together. Abductive networks are applied to automatically learn the empirical functional relationships between the data sensed on the other spectral channels to calculate the data that should have been sensed on the corrupted channel. Using imagery unaffected by noise, it is demonstrated that abductive networks correctly predict the noise-free observed data.

1 Introduction.

The fleet of four polar orbiting meteorological satellites currently operated by the National Atmospheric and Oceanic Administration (NOAA) carries a multi-spectral sensing system for imaging the Earth. This system, the Advanced Very High Resolution Radiometer (AVHRR), measures irradiances in five narrow spectral bands ranging from the visible to the infrared (IR) parts of the electromagnetic spectrum. The system is described in section 2 below. Suffice it to say here that by virtue of the high resolution of the instrument, a wealth of data is available.

It has been noted that one of the five spectral channels of the AVHRR (channel 3) is particularly susceptible to noise and its accuracy degrades with age, perhaps to the point where the data is unusable (Ref. 1). The possibility also exists that some of the archived AVHRR imagery from the older satellites that have been replaced with the current generation of spacecraft may also be of questionable quality.

The problem faced is the use of archived and real-time satellite imagery which may be partially corrupted by noise. One approach is to correct the data to its true but *a priori* unknown value. Because the channel is continually and progressively denigrated by noise, any correction scheme requires constant maintenance.

¹ SAIC Technical Report SAIC-94/1062

An alternative approach, pursued in study described here, is to replace the data measured on the noisy channel with data constructed from the other four spectral channels. Our approach relies on a technique called abductive networks that automatically discovers the networking between the spectral channels that are embedded in the measured data. In this way the noisy satellite imagery is reconstructed according to empirical functional relationships that tie the spectral channels together.

Here we describe the application of a proprietary tool for creating abductive networks to the modelling of the AVHRR. Specifically, channel 3 is modelled as the output calculated from the empirical inputs of the other four spectral channels. Our approach was exercised on imagery collected with the AVHRR on NOAA-11, which is not as yet seriously compromised by noise. The data predicted for channel 3 with the other channels as inputs to the network that was created is then statistically compared to the data actually observed. The result was that the network was highly successful at simulating the observed output.

The next section provides a short description of the AVHRR. Section 3 gives an overview of abductive technology. Section 4 describes the application of abductive networks to satellite imagery with the objective of uncovering the effective relationship between the imagery sensed in an intermediate spectral band and the imagery sensed in the neighboring bands. Our conclusions, principally that abductive networks show great promise for reconstructing noisy satellite imagery, are presented in section 6.

2 A Brief Description of the AVHRR.

The AVHRR currently flown aboard the NOAA polar orbiting meteorological satellites is a downward-pointing cross-track scanning system. It makes radiometric measurements in five spectral channels: two in the visible and adjacent near-infrared (near-IR) part of the spectrum (channels 1 and 2) and three in the IR part (channels 3, 4, and 5). The spectral band widths, in microns (μm), are summarized in Table 1. For NOAA-10 only, the spectral band of channel 4 is 10.50 - 11.50 μm , and channel 5 output is a repeat of channel 4. The field of view of each channel is approximately 1.4 milliradians leading to a nadir resolution of about 1.1 km (for a nominal satellite altitude of 833 km). There are 2048 pixels per scan line, where each pixel covers about 2 steradians.

Table 1. Spectral band widths of the AVHRR.

Channel #	Band Width (μm)
1	0.580 - 0.680
2	0.725 - 1.100
3	3.550 - 3.930
4	10.300 - 11.300
5	11.500 - 12.500

The analogue data output from the sensors is digitized on board the satellite. The IR channels are calibrated in flight using a view of a stable black body and space as a reference. No in flight visible channel calibration is performed, although the space view is available as a reference point.

The radiometer data collected by channel 3 of each NOAA satellite have been very noisy due to sensor problems and may be eventually unusable (Ref. 1). This is especially true when the satellite is in daylight. (Of course, channels 1 and 2 are blank for nocturnal views.)

The normal operating mode of the AVHRR scanning system is to capture a scan line in a buffer and continuously broadcast the digital data in a wide beam aimed at the Earth. The direct transmission mode is called High Resolution Picture Transmission (HRPT). Ground processing of the HRPT consists of its calibration, earth location, and breakout of the individual sensor channels.

3 An Introduction to Abductory Induction and Abductive Networks.

Abductive reasoning, or abduction, is defined as the process of reasoning under conditions of uncertainty from general principles and initial facts to new facts (Ref. 2). Abduction differs from deduction, in which all principles and facts are assumed to be known with complete or assumed certainty.

Induction is the process of reasoning from specific facts to general principles. This reasoning process handles the many real-world situations that are rich in empirical data but lack sufficient conceptual understanding to unify that data into a coherent, accurate view of the world. Ideally, the facts supplied to an inductive argument are known with absolute certainty. In the real world, however, the facts are contaminated with uncertainties. Uncertainty arises, for example, from imprecise, unreliable or incomplete information. Even with indisputable information, uncertainty arises due to a lack of complete and thorough knowledge and understanding of the situation. Then the generalities inductively reasoned from those facts must themselves be uncertain. As a result, the reasoning itself contains uncertainty. Abduction is the reasoning process that incorporates this realistic view of uncertainty.

A practical implementation of abductive reasoning uses numeric functions and measures, called abductive measures, to convey the inherent importance of a single fact or piece of information (Ref. 2). Abductive measures represent relationships between facts. These should be viewed as working, rather than 'true', relationships in the sense that they predict nature correctly even if for the wrong reasons.

Abductive measures are used to decompose complex problems into subproblems in a process called chunking. Here a limited number of facts, or types of facts, are dealt with at a time. They are summarized in terms of single abductive measures. The chunks are then united by appropriately combining their respective abductive measures.

Abductory induction is the process of creating general principles from databases of empirical observations. Abductory induction is applied to create an abductory model of the process

described implicitly by the database by formulating, or at least approximating, the relationship between the database variables in terms of the contained data. The abductive model is most conveniently posed as an unstructured network or a cascade of mathematical equations. This adaptability property makes abductive induction particularly well suited to unsupervised machine learning. The problem immediately posed is determining the appropriate functions, and hence the layout of the network, out of an infinite number of candidates, that best describes the data. Assuming a model structure, as is done in regression techniques, may result in poor fits to the data just because that specific structure is not present. An alternative is to model the data with a very general multinomial. Within very broad assumptions any arbitrary function may be approximated by a polynomial (i.e., a truncated Taylor series); the accuracy of the approximation is directly related to the number of terms retained, that is, the degree of the polynomial. However the many coefficients needed for even a small set of variables makes this approach intractable for degrees much above 2.

A practical solution to modelling the database in terms of its uncertain structure is to apply the chunking concept and split the input variables among several groups. The groups are collectively input to the individual nodes of an incipient network and the relations among them are summarized in terms of an abductive measure. These results are then passed on to the next layer of the evolving network. The labor is substantially reduced because only the model associated with a single node must be determined at any single time.

Abductive networks are networks of functional nodes (Ref. 3). Neural networks may be considered a special class of biologically-motivated abductive networks. Incorporating the chunking concept, a very effective algorithm for creating abductive networks utilizes polynomial equations (of moderate order) for the abductive measures. Given a database of example situations about a problem consisting of a representative set of inputs and outputs, an abductive network can be used to fit the best polynomials relating the variables, node by node, cascading layer to layer. Specifically, inputs to each node are processed and output, along with the original input variables to the nodes in subsequent layers of the network. The result is a compact representation of the interactions between the variables as evidenced in the massive amount of empirical data.

The Abductive Induction Mechanism (AIM™) is proprietary software of AbTech Corporation for implementing abductive induction for the automatic and unsupervised creation of abductive networks (Ref. 4). The network created by AIM is a robust and efficient representation of the relationships existing between the variables contained in the database. AIM uses polynomials of up to degree 3; the polynomials contain cross-terms to allow interaction between node inputs. Not all terms may be included in specific nodal polynomials because AIM, in a process called carving, neglects terms which do not contribute significantly. The network size, chunking and connectivity (between chunks and/or inputs), and coefficient values are all determined automatically by AIM. Networks are created from layer to layer until the network model ceases to be improved according to a modeling criterion. The criterion assures that as accurate a network as possible is created without overfitting the data (that is, tailoring the network specifically to the supplied database).

4 An Application of AIM to the AVHRR Calibrated Channel Output.

It was mentioned above that, for the older satellites in the NOAA fleet, channel 3 is very noisy, to the point of being unusable without significant suppression of the noise effects. The objective is to reconstruct channel 3 from the other four spectral channels.

The AVHRR instrument scans the scene pixel by pixel in all five spectral channels simultaneously. Of course, the sensors will make different irradiance measurements in the different spectral bands. However, in a pixel, excluding any possible misalignment among the five fields of view, the channel 3 irradiance must be related to the irradiances measured on the other spectral channels. That relation is a complex problem in radiative transfer for both solar and terrestrial photons. An alternative to a possibly intractable theoretical analysis is to use a satellite imagery database consisting of AVHRR calibrated output to uncover empirical relationships between channel 3 and the other channels contained in the data.

The SAIC satellite ground station at received imagery from the NOAA-11 satellite for a pass over the eastern United States on 25 February 1994 around 2139 UTC (16:39 EST). NOAA-11 was launched September 24, 1988; as the second oldest satellite in current operation, it is two years older than NOAA-10 and nearly four years older than NOAA-9. The AVHRR on NOAA-11 has not yet evidenced severe denigration of any of its spectral irradiance measurements. The downlinked data was calibrated, rectified, and broken out into its individual channels, which were separately saved to file. The satellite image contained in excess of 1000 scan lines. A 500 line by 500 pixel box was extracted from the southwest corner of the image and sampled for every other scan line, so that the channel databases each contained 250 scan lines nominally separated by 2.2 km in the direction of the satellite track. The data for each individual channel were then ordered by pixel, for a total of 125000 pixels, in a single file for each of the five channels. Each pixel is considered as an individual observation containing five values, one for each of the spectral channels.

The AIM software package was applied to the image box. Memory limitations in AIM prevented use of the entire database because AIM is limited to only 8000 observations. As a result, four 8000-pixel strips were extracted from the image box. Specifically, the extracted image box was divided into four sections in the along-track direction. Each section of the image box was sampled in blocks of 8000 pixels such that each block contains sixteen sequential 500-pixel neighboring scan lines; the blocks are spatially coherent. A fifth block was created by assembling four adjacent scan lines from each of the four image strip. Note that while the quarters of this image block are spatially coherent, the quarters are spatially decoupled from each other.

Channels 1, 2, 4, and 5 were designated as the network inputs and channel 3 was designated as the network output. Individual networks were created for each of the five image blocks. The networks were created on a Macintosh SE/30. The time required for forming the network obviously depended on network complexity. Creation times ranged from about 20 minutes up to nearly an hour. Each network was then evaluated against both its own creating image block and the other four image blocks. The evaluation consisted of statistically comparing the channel 3

output predicted by the network to the channel 3 data actually observed.

All of the networks performed well on self-evaluation. Generally the networks degraded with spatial distance of the evaluating image block from the network-forming image block, that is, with progressive spatial decoupling between the image blocks. The exception was the AIM network created with image block 5 (the four-strip composite through the 500 line by 500 pixel image box). That network, presented in figure 1, generally outperformed all the other networks, except for their own self-evaluation.

As can be seen in figure 1, the network created with image block 5 is a four-layer network of feed-forward elements, that is, the network cascades from the raw input variables on the left to the single output variable on the right. The inputs are the calibrated AVHRR data from channels 1, 2, 4, and 5. Note however that only channels 1, 4, and 5, referred to as ch1, ch4, and ch5, respectively, were used. The final output is the network-predicted (calibrated) response for AVHRR channel 3, referred to as ch3. Channel 2 was carved from the inputs because of its partial redundancy, most likely with channel 1. The numbers and types of network elements, the element polynomial functions, and their connectivity are learned abductively (induction under uncertainty). The coefficients of the element functions are determined by multiple linear regression of terms up to power three. The structure of the network is determined according to a set of rules and heuristics that are an inherent part of the AIM network creation strategy. The best network, in terms of its structure, element types, coefficients, and connectivity, is found automatically by minimizing a modeling criterion that seeks the most accurate network possible within acceptable tolerance (this avoids creating a network tailored to only the training data).

In figure 1, the open circles following the inputs are 'normalizers'. They transform the original input variables to standard variables with zero mean and unit variance. This assures that all input variables will be fairly represented in the network. The boxes labelled double and triple are elements whose name is based on the number of inputs from the previous layer. These elements are described by fairly general third-order polynomials. Doubles and triples may have some significant explicit cross-product terms, allowing interaction among the node input variables. Note that the output of any given element can feed subsequent layers as can the original variables. The open circle preceding the network output (ch3) is a so-called 'unitizer'. A unitizer converts the standardized range of the intermediate network output to the units of the output variable used to create the network; it is an inverse normalizer.

Figure 2 plots the observed output of channel 3 in image block 5 against the output predicted for channel 3 using the data measured on the other channels in block 5 as input to the network created with block 5. This is a self-evaluation of the network created with image block 5. The line with unit positive slope indicates perfect correlation between the observed and the predicted channel 3 output. The overwhelming bulk of the 8000 network-predicted channel 3 values straddle the line, indicating the high quality of the network fit to the observations. The correlated data appear to group predominantly into two large clusters hugging the unit line (the upslope cluster being the more massive of the two). Apparently the observed channel 3 data is inherently bimodal; this bimodal distribution is captured in the network predictions. Figure 3 displays the normalized errors for this self evaluation of the block 5 network. Normalized error is defined as

the difference between the observed and predicted values for the channel 3, normalized by the observed value. As in the previous figure, the normalized errors group predominantly into two large clusters hugging the zero-error line. The larger of the two clusters sits over the mean of the channel 3 observations for block 5 (2324.3). The normalized errors are mainly within $\pm 5\%$, and nearly evenly dispersed around the zero-error line. As the observed values depart from the block mean, error grows, implying that the network performance degrades. Even at its worst, however, the normalized error is mostly within about 15%.

Figure 4 plots the observed output of channel 3 in image block 3 against the output predicted for channel 3 using the data measured on the other channels in block 3 as input to the network created with block 5. The perfect correlation line with unit positive slope is displayed for comparison. Here, the agreement between the predicted and the observed values is excellent, as evidenced by the near perfect collapse onto the 45° line. The normalized errors between the block 5 network predictions of channel 3 for block 3 and the actual block 3 observations are shown in figure 5. The normalized errors are strongly clustered about the actual block mean of 2316.9, and are about 1%. Note that in general the network tends to very slightly overpredict the channel 3 output. The strong clustering of the normalized errors reflects the shorter range and tighter clustering of the block 3 data about its mean (standard deviation = $27.0 \sim 1.2\%$ of the mean).

5 Conclusions.

We have demonstrated that that abductive networks are very successful in modelling the measurements collected with the AVHRR in our specific test case. The abductive networks created with AIM create reliable and compact representations of the AVHRR spectral channels in terms of diagnosing the empirical relationship between channel 3 and the other four spectral channels. The network trained with the composite database selectively extracted from the imagery so as to have only partial spatial coherence generally outperformed the networks trained with spatially coherent databases, except perhaps for the self-evaluation. This indicates some near-universality exists in the relationship between the channels, which may be found by the appropriate sampling of the satellite imagery. The general use of abductive networks for modelling the AVHRR towards reconstructing the noisy data collected on its channel 3 shows great promise.

Another possible use for abductive networks is as a quality-control monitor. Specifically, the real-time degradation of channel 3 can be measured by periodically comparing its observations to network predictions. For example, the channel may be considered corrupted if the average error, say, between the observed and the predicted channel 3 output through an image (or a piece of an image) exceeds some established threshold.

6 References.

1. Kidwell, K.B. (ed.), NOAA Polar Orbiter Data Users Guide, National Environmental Satellite, Data, and Information Service, Satellite Data Services Division, Washington, D.C., July 1991.

2. Montgomery, G.J. and Drake, K.C., *Abductive networks*, SPIE Applications of Artificial Neural Networks Conference, Orlando, FL, 16-20 April 1990.
3. Montgomery, G.J., Drake, K.C., and Montgomery, K.M., *Real-time diagnostics using abductive machine learning*, Ninth Annual Control Engineering Conference, Rosemont, IL, 22-24 May 1990.
4. AIM Users Manual, Macintosh version 1.0, AbTech Corporation, Charlottesville, VA, 1990.

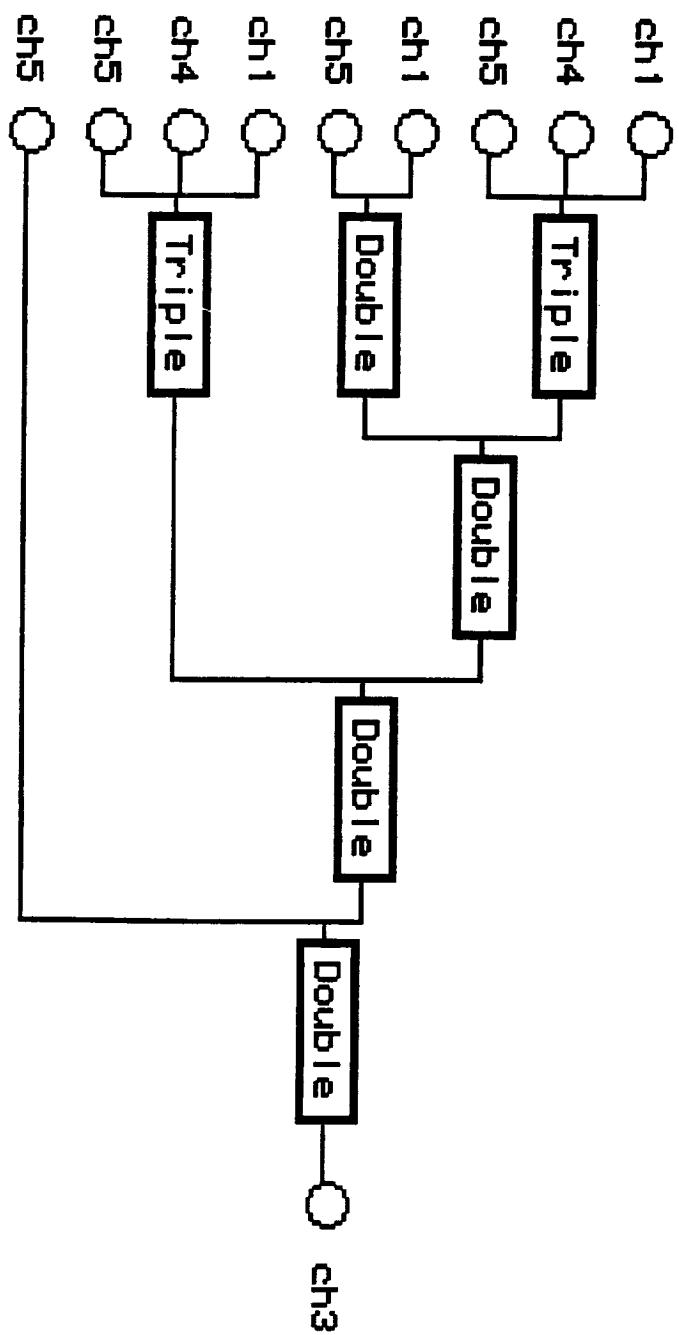


Figure 1. The AIM abductive network created with image block 5 (the composite block through the image box) showing the numbers and types of network elements.

Predicted Channel 3 Output

Observed Channel 3 Output

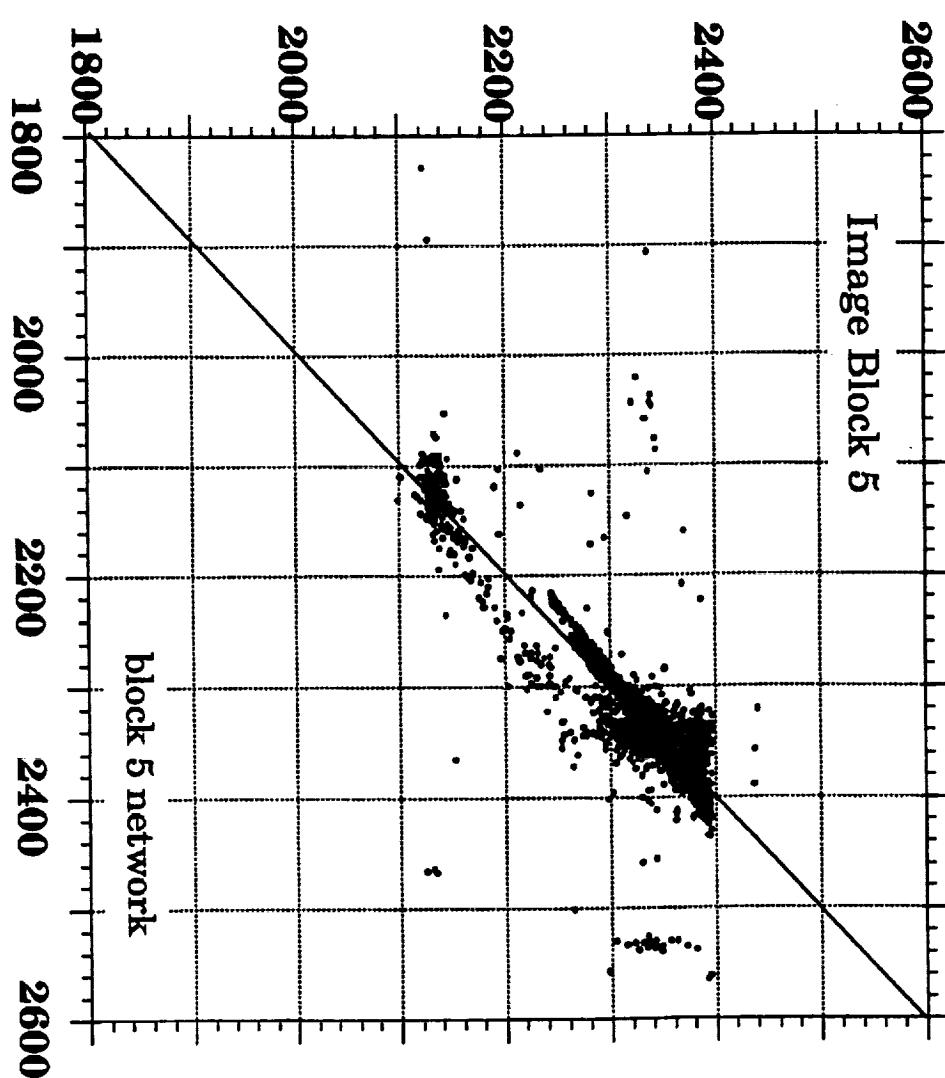


Figure 2. Observed channel 3 output in image block 5 vs. the predicted output using the other block 5 channels as inputs to the network created with image block 5.

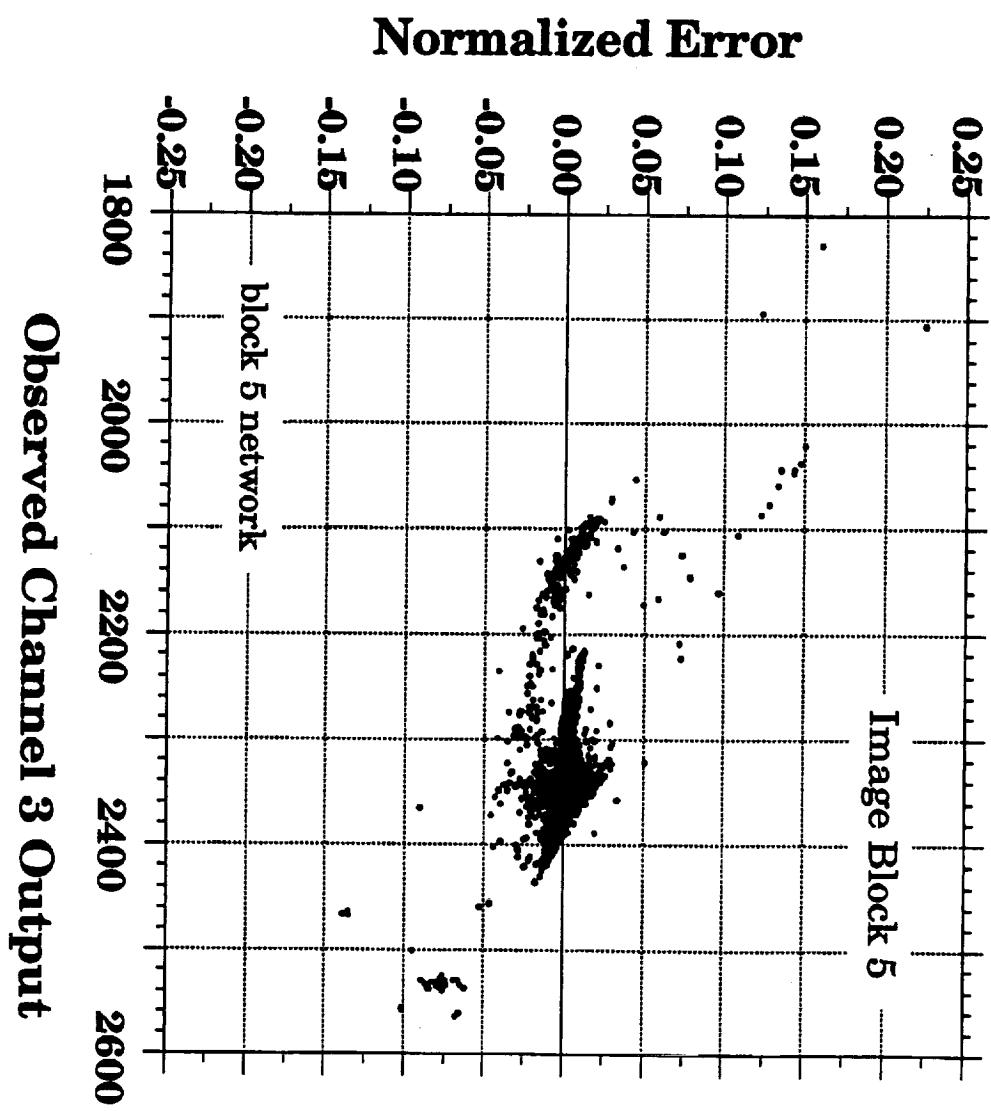


Figure 3. Normalized channel 3 output errors in image block 5, between the observed output and the output predicted from the network created with image block 5.

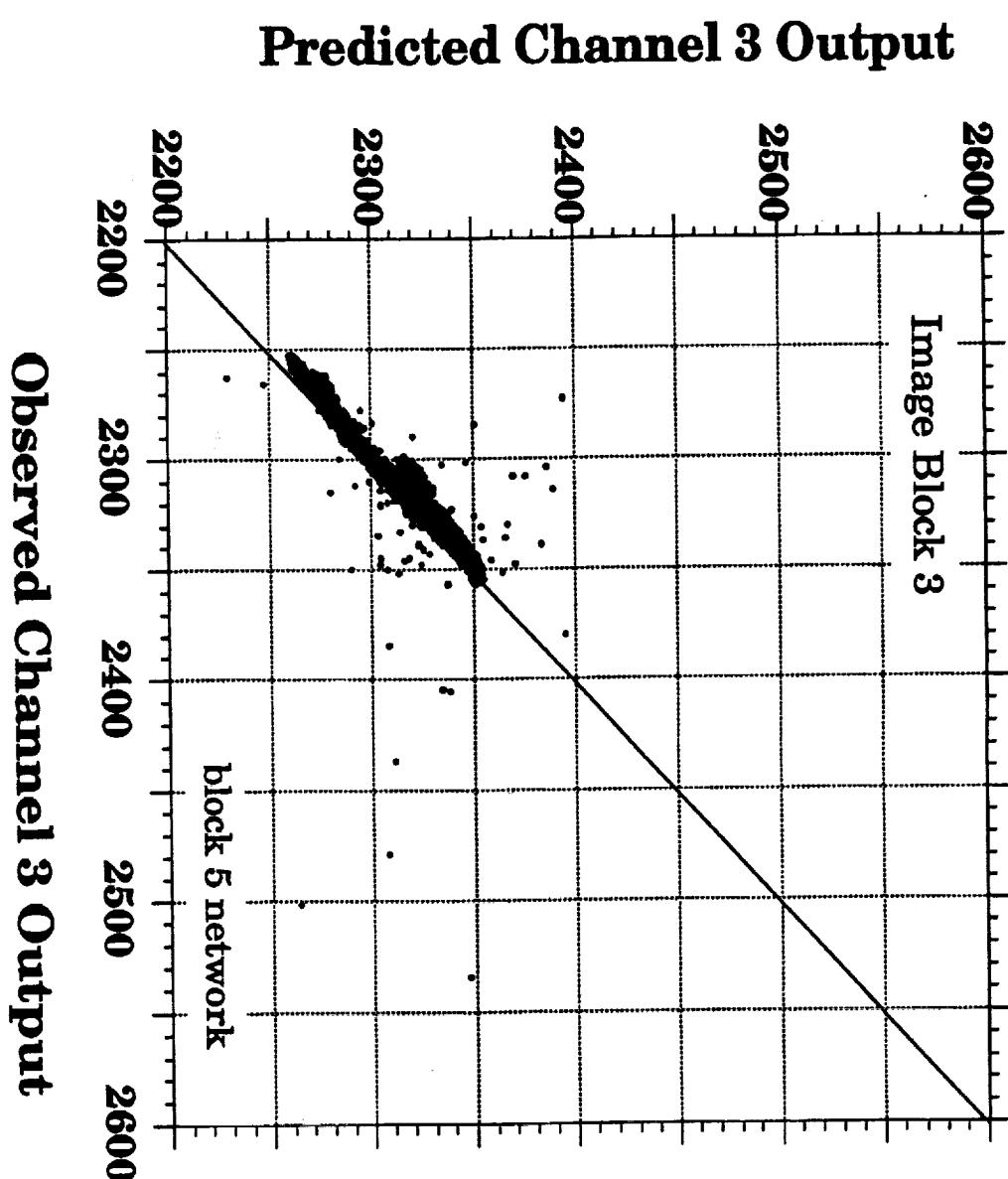


Figure 4. Observed channel 3 output in image block 3 vs. the predicted output using the other block 3 channels as inputs to the network created with image block 5.

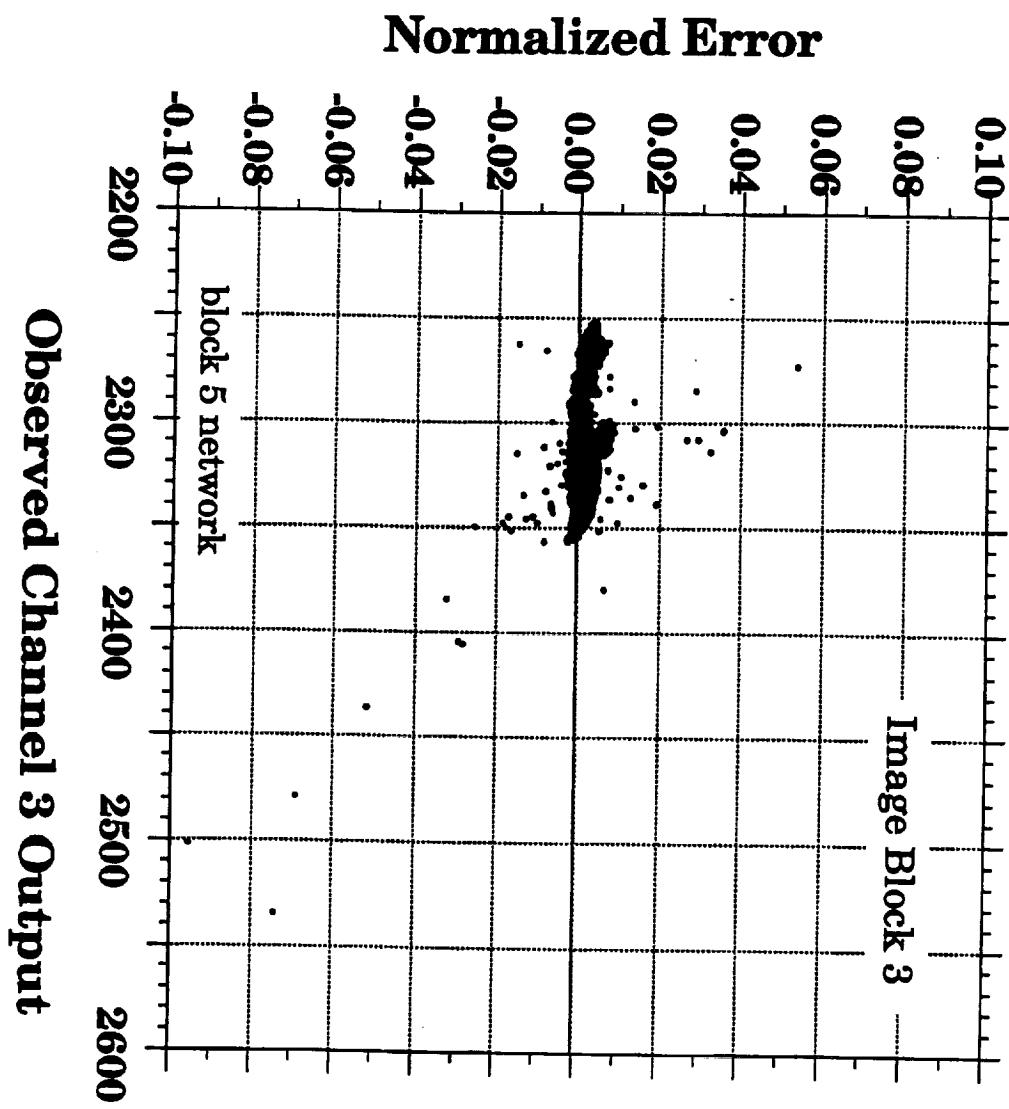


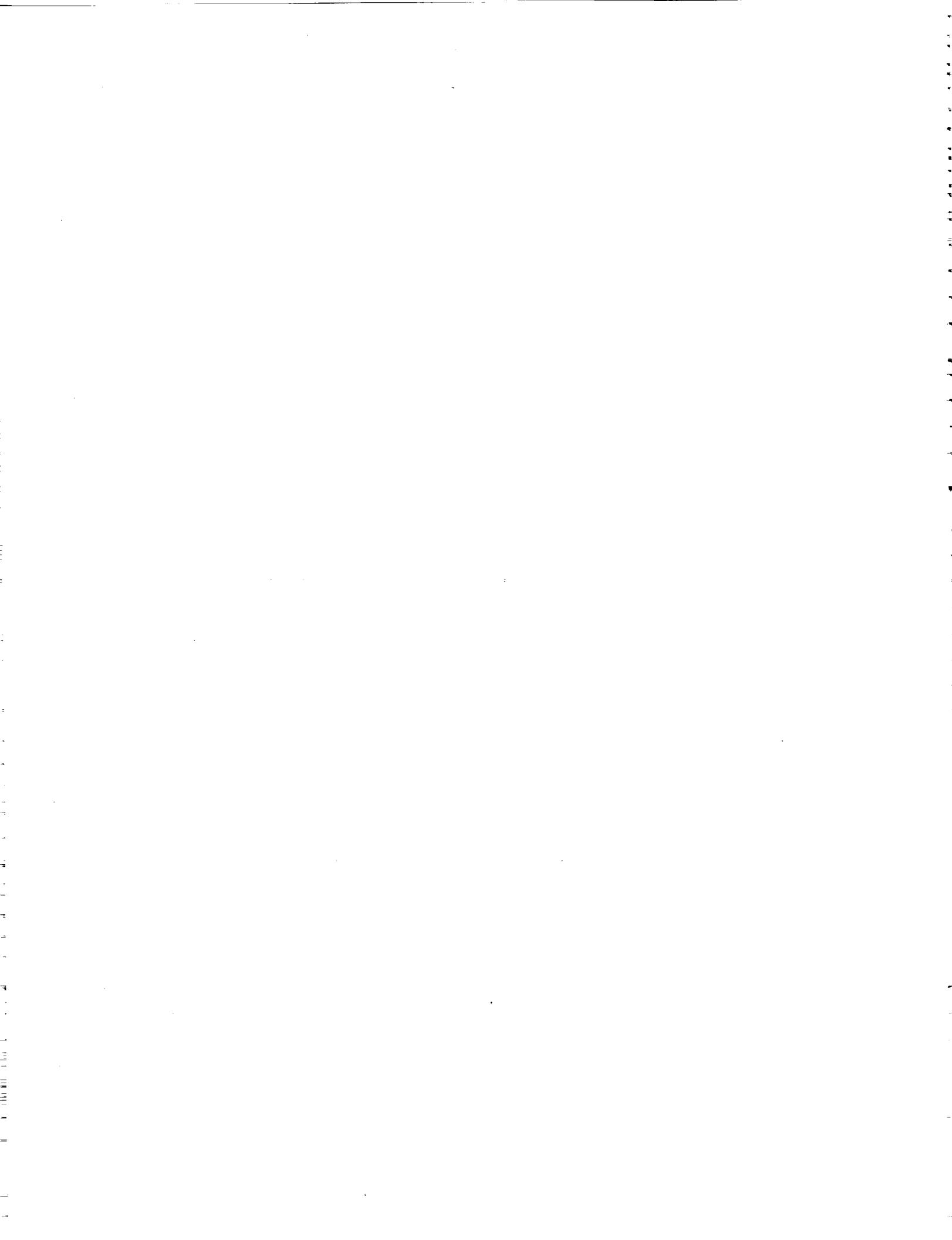
Figure 5. Normalized channel 3 output errors in image block 5, between the observed output and the output predicted from the network created with image block 5.



On/T

System Monitoring and Diagnosis

PRECEDING PAGE BLANK NOT FILMED



1995120969

N95-27390

S,5-19

47265

p. 13

Model-based monitoring and diagnosis of a satellite-based instrument.

André Bos,* Jörg Callies, and Alain Lefebvre.[†]

Abstract

For about a decade model-based reasoning has been propounded by a number of researchers. Maybe one of most convincing arguments in favor of this kind of reasoning has been given by Davis in his paper on diagnosis from first principles (Davis 1984). Following their guidelines we have developed a system to verify the behavior of a satellite-based instrument GOME (which will be measuring Ozone concentrations in the near future (1995)). We start by giving a description of model-based monitoring. Besides recognizing that something is wrong, we also like to find the cause for misbehaving automatically. Therefore, we show how the monitoring technique can be extended to model-based diagnosis.

1 Introduction

1.1 Testing complex systems

Before space systems, like satellite-based instruments, go into orbit, it is important to validate the system's functioning thoroughly. However, as systems become more and more complex, the effort needed to verify these systems becomes enormous. Traditional testing methods validate system behavior by applying test inputs and comparing observed to expected output behavior. Care must be taken that all possible interactions between subsystems are covered. Unfortunately, experience shows that it is nearly impossible to do complete testing, and most systems possess some unknown –and unwanted– behavior. In these cases it is very important to know if the system (*e.g.*, when it is in orbit) behaves correctly. For example, a faulted component of an Ozone measuring instrument may influence the measurements negatively. So, it is important to recognize malfunctioning as soon as possible. However, for a human controller it is just impossible

*TNO-TPD-TU, Instrumentation Department, P.O.Box 155, 2600 AD Delft, The Netherlands. bos-a@tpd.tno.nl.

[†]ESTEC-ESA, ERS Project Division, Earth Observation Department, P.O.Box 299, 2200 AG Noordwijk, The Netherlands. {jcallies,alefebvr}@estec.esa.nl.

to monitor system's performance in every detail. An automatic system is needed to keep track of the system.

In this paper we describe a technique to validate systems: *model-based monitoring and diagnosis*. Note that there is no intent to replace existing test techniques; it is an *additional* method that is used to detect the errors that remain after traditional testing and when the system is in operation. The test method here presented is model-based. That is to say, a behavior description is used to predict how the system should behave, and the predictions are compared to the actual observations. If an inconsistency arises, then it is assumed that something is wrong and an error is signaled.

1.2 GOME

We have applied the test method to verify the GOME instrument (ESA 1993). GOME, short for Global Ozone Monitoring Experiment, is an instrument that will be mounted on ESA's ERS-2 satellite. Its purpose is to measure Ozone concentrations in the earth's atmosphere. This is done by comparing the sun's spectrum measured directly to the spectrum of sun light that has been reflected and travelled twice through the earth's atmosphere.

Apart from a diode array for measuring the spectra, the instrument has a number of supporting subsystems. Such as a command interpreter for interpreting and executing of commands send by ground control; a data acquisition unit for sending the measured spectrum and house keeping data to ground control; a mirror unit for scanning the earth's atmosphere; a heating unit for temperature control; etc.. All in all, GOME is a rather complicated system and its behavior is hard to verify.

1.3 Overview of the paper

In Section 2, we start by describing a monitoring system that is used to verify GOME's behavior. A monitoring system checks if a system is functioning correctly, however, the cause of a malfunctioning is not reported. This is part of the functionality of a diagnostic system. In Section 3, we extend the description to a diagnostic system that is currently being implemented for GOME. Finally, in Section 4, some conclusions are drawn and future work is described.

2 Monitoring

As already described we have implemented a *model-based* form of monitoring. We assume that something is wrong whenever the model's predictions are contradicting the observations of the system's behavior. That is, a description of normative behavior is used to verify the system.

In this section we will (1) formalize the method of model-based monitoring in a way so that it is easily extended to model-based diagnosis, see Section 3; (2) describe the implementation of it; and (3) discuss some of the results of applying a monitor program to the GOME instrument.

2.1 Characterizing model-based monitoring

In this section we will establish a conceptual framework for defining model-based monitoring. Central in this framework, and also in that for diagnosis, is that we view a description of system behavior (a model) as a formal system. That is, the behavior description is a set of sentences taken from some kind of language with a logic attached. It is important to note that we do not restrict ourselves to predicate or first-order logics. To the contrary, we view formal systems in which algebraic or differential equations can be expressed as important candidates for logics in which the behavior of a system can be expressed. Viewing the behavior description as a formal system eases the definition and implementation of monitoring and diagnostic system, but may also introduce notations that may seem awkward in the context of system theory. For example, a numerical integration step is –in the logical context– considered as a derivation rule, *e.g.* Euler's can be stated as:

$$\begin{aligned} x(t) &= C_1, && \text{with } C_1 \in \mathbf{R}^n, \\ \underline{x'(t) = Ax(t)} \\ \underline{x(t+1) = AC_1 + C_1}, \end{aligned}$$

with $x(t) \in \mathbf{R}^n$ and $A \in \mathbf{R}^n \times \mathbf{R}^n$. The derivation of σ from a set Σ is denoted as:

$$\Sigma \vdash \sigma.$$

Consider for example the case of dynamical simulation. Let *SIMMOD* denote a dynamical simulation model and *INIT* its initial conditions both expressed in some formal system with Euler's integration step as derivation rule. Then the set

$$PRED = \{p : SIMMOD \cup INIT \vdash p\}$$

contains all the predictions that can be obtained by applying the derivation rules of the formal system.

Using the logical terminology, we define a system to be monitored as follows:

Definition 2.1 *A system to be monitored is a triple $(OBS, MODULES, SD_m)$, where*

- *OBS, the observations, is a finite set of observations each of the form*

$$v = \langle value \rangle,$$

where v is a variable, and $\langle value \rangle$ a value of appropriate type.

- *MODULES*, the modules, is a finite set of so-called modules. Modules are introduced to denote subsystems that are supposed to be functioning independently. For each module a separate set of behavior relations is defined, as will be explained next.
- SD_m , the system description (for monitoring), is a finite consistent set of behavior relations for each module. The general form of a modular behavior specification is as follows:

$$M \supset \langle \text{behavior relations for } M \rangle,$$

where $M \in \text{MODULES}$ and \supset denotes material implication (if ..., then ...).

Due to a possible incomplete knowledge of the system, e.g. the current state is not known, we allow alternative behavioral relations per module; however, exactly one of these behavioral relations must be true. That is, each instance of “⟨behavior relations for M ⟩” is of the form:

$$\text{rel}_1 \oplus \cdots \oplus \text{rel}_n,$$

where rel_i is e.g. an algebraic or differential equation, and $\sigma_1 \oplus \sigma_2$ denotes the fact that either σ_1 or σ_2 is true, but not both¹. \oplus is also called a choice operator.

When monitoring a system, an error message must be generated whenever the predictions made by the system description are contradicting the observations. A contradiction occurs whenever a prediction assigns a value to a variable that is incompatible to the observations². Deciding whether two values are incompatible is problem and type dependent. For example, for real-valued variables normally a range on the values is defined; for variables with a discrete domain the values have to match exactly. Furthermore, because the different modules are assumed to be working independently, we can give an indication where something is going wrong by stating the module responsible for generating the contradiction. This leads to the following.

Definition 2.2 Let $(OBS, \text{MODULES}, SD_m)$ be a system to be monitored. An error message for a module $M \in \text{MODULES}$ is generated whenever

$$SD_m \cup OBS \cup \{M\}$$

is inconsistent³.

¹ $\sigma_1 \oplus \sigma_2$ is an abbreviation for $\sigma_1 \vee \neg\sigma_2$, and $\neg\sigma_1 \vee \sigma_2$.

²Because SD_m is assumed to be consistent, contradictions may only occur due to a mismatch between prediction and observations.

³Note that presence of M in the formula enables the use of its behavior relations.

It is important to note that we assume that only one module –and no combination of modules– is responsible for a contradiction. In other words, multiple faults (de Kleer and Williams 1987) are not captured by this definition (this will be handled in the section on diagnosis, see Section 3). Note, however, that during a monitoring session more than one module may generate an error message. If we recall the initial purpose of the monitoring system, *viz.* the verification if a system functions correctly, the restriction to single faults is not that serious. We assume that the modules are chosen so that one module captures the behavior of a subset of the system constituents. On the occurrence of an inconsistency, we know that the culprit is to be found within that subset.

2.2 Implementation

We have implemented a monitoring system to verify GOME's behavior. In Figure 1 the overall layout of the program is given. To simplify the implementation,

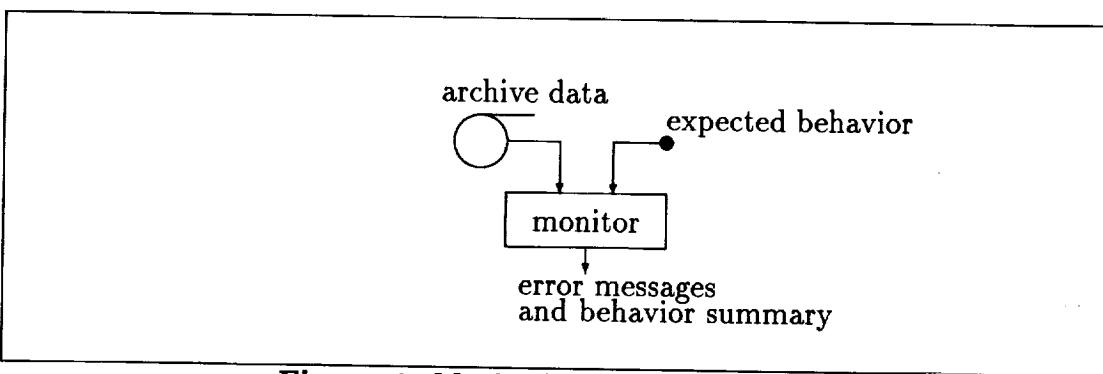


Figure 1: Monitoring system for GOME

the observations (OBS) of GOME's behavior are first stored on (Bernoulli) disks before GOME's operation is analyzed. A snapshot (the values of all GOME's variables) is taken each 1.5 secs. and is stored in what is called *archive data*. The contents of a single snapshot is called a *packet*. *Packet numbers* are used to address packets.

The *expected behavior* comprises the system description per module⁴. SD_m can be considered as a kind of simulation model of the system where the behavior relations are centered around the modules. Note that SD_m is not truly a simulation model because the choice operator introduces alternative behaviors per module. So, no conclusive predictions can be made using SD_m ; it can only be used to do a consistency check.

⁴In the current implementation the program and the system description is coded in C (Kernighan and Ritchie 1978). The behavioral relations are decoded as procedures; a more elegant –at least viewed from a logical and a maintenance perspective– implementation would use a declarative description of both the behavior and the derivation (*e.g.* Euler's rule) relations.

The *monitor* program reads each packet from the archive data (OBS), ‘simulates’ each module M using the expected behavior (SD_m) and checks if

$$SD_m \cup OBS \cup \{M\}$$

is inconsistent. If so, an error message with some additional information is printed.

The following example gives some feeling for the implementation of the monitoring program.

Example 2.1: Consider the operation of the setting of the mirror’s mode. Informally, SD_m contains relations that describe the following behavior:

If a command that sets the mirror in swath mode is in the current packet, then after N packets⁵ ($= N \times 1.5$ secs.) the mirror position is changing according to a linear relation defined on the packet number.

Now, if a mirror-setting command is found in the current packet, the monitor program checks after a delay of N packets the mirror position. \square

As an extra, the monitor program prints for each packet –what we call– a *behavior summary* with the most important status information of GOME’s operation. For example, the behavior summary contains the last submitted command, the mirror’s and coolers’ mode, and so on. This extra information comes in handy when the cause for malfunctioning is searched (either manually or automatically with a diagnostic system).

2.3 Results

The monitoring system as described above has been applied to the GOME instrument. It should be clear that a monitoring instrument does not perform a full functional test. Types of behavior that are not enabled during the verification process will not be tested for correct functioning. As we have already mentioned, it is an additional method of testing. Although GOME was tested fairly intensively, the monitoring program did expose a number of faults. To give some feeling for the type of faults, we name a few: (1) The integration time (for measuring sun light) was set incorrectly on a number of occasions; (2) synchronization faults of timers on receipt of a command; (3) a too slow operating timer; (4) inaccurate scan mirror positioning during swath mode; (5) documentation faults (other process variables are measured than documented); (6) etc..

3 Diagnosing

The monitoring program has been proved to be useful for validating the correct functioning of GOME. However, when an error message is generated, the cause

⁵Actually, this number N depends on the current packet number modulo 4.

for misbehaving has to be searched manually. It is interesting to have a system that not only recognizes that something is wrong, but also is able to find the cause of misbehaving. This functionality is part of *diagnostic* systems.

3.1 Characterization

Similar to model-based monitoring, the characterization of model-based diagnosis uses a logical terminology, see *e.g.* (Reiter 1987; de Kleer and Williams 1987; de Kleer and Williams 1989).

Definition 3.1 *A system to be diagnosed is –again– a triple $(OBS, COMP, SD)$, where*

- OBS , a finite set of observations, defined as in the case of monitoring.
- $COMP$, a finite set of components. Components are akin to modules, however, behavior is assigned to individual components. In this way, it is possible to extract responsible components for a discrepancy in observed and expected behavior.
- SD , the system description (for diagnosis), similar to the model-based monitoring case, except that the behavior relations are defined per component.

In the diagnostic case we assume that a component working in a mode. A mode represents a physical ‘condition’ (so to speak) of a component. For example, we have:

- A normal mode, i.e. the component is working as intended.
- One or more fault modes, i.e. the –faulted– component is working according to a known behavioral relation.
- An abnormal mode, i.e. the component is not working as intended but we have not anticipated its fault behavior as in the previous case.

Now, the general form of a behavior relation is:

$$\text{Mode}(c) \supset \langle \text{governing eq.} \rangle,$$

where $c \in COMP$, and “⟨governing eq.⟩” describes how the component’s variables are governed when c is working in mode $\text{Mode}(c)$. If the $\text{Mode}(c)$ is the abnormal mode, then the equation is such that no predictions can be made.

To each mode of a component a prior probability is assigned. This prior probability is used during the computation of diagnoses as will be explained shortly.

SD can be considered as a component-centered simulation model. That is, behavior relations are given per component, so components responsible for a discrepancy can be isolated. In GOME's case we have the following.

Example 3.1: We consider two components: c_m and c_c representing the mirror unit and command interpreter, respectively. A normal functioning mirror unit (c_m) can scan the earth's atmosphere either at a fixed position or rotating:

- At a fixed position, indicated by the predicate $fixed(c_m, t)$ being true for all time instances t the mirror is fixed. The position of the mirror at time instance t has a constant value: $pos(t) = m_k$ ⁶.
- With a scan angle, indicated by the predicate $swath(c_m, t)$ being true for all time instances t the mirror is rotating. The position of the mirror at time instance t has a value that is linearly dependent on t described by the function $f(t)$ ⁷.

In Figure 2 the behavior of the mirror is given.

$$Normal(c_m) \supset \left[\begin{array}{l} fixed(c_m, t) \supset pos(t) = m_k \\ \vee \\ swath(c_m, t) \supset pos(t) = f(t) \\ \vdots \end{array} \right]$$

Figure 2: Mirror unit behavior

The command interpreter c_c sets, among other things, the predicates $fixed(c_m, t)$ and $swath(c_m, t)$ if a corresponding command has been received⁸, see Figure 3. \square

Now a *diagnosis* is an assignment of modes such that no predictions can be made that are contradictory to the observations. We use the following definitions.

Definition 3.2 A mode assignment is a conjunction of mode predicates for all $c \in COMP$:

$$\bigwedge_{c \in COMP} Mode_c(c).$$

⁶This is simplified, actually the position can be controlled.

⁷Again this is simplified; it is possible to control the maximum angle of rotation.

⁸It is assumed that once the predicate $fixed(c_m, t)$ or $swath(c_m, t)$ is believed, it stays true until it is explicitly asserted false.

$$Normal(c_c) \supset \left[\begin{array}{l} cmnd = fixed_{mirror} \supset fixed(c_m, t + N) \wedge \\ \quad \neg swath(c_m, t + N) \wedge \\ \quad \dots \\ \vee \\ cmnd = swath_{mirror} \supset swath(c_m, t + N) \wedge \\ \quad \neg fixed(c_m, t + N) \wedge \\ \quad \dots \\ \vdots \end{array} \right]$$

Note that N denotes the delay after which a scan mirror command is issued.

Figure 3: Command interpreter behavior

Definition 3.3 A diagnosis for a triple $(OBS, COMP, SD)$ is a mode assignment Γ such that:

$$SD \cup OBS \cup \{\Gamma\}$$

is consistent.

Recall that assuming a mode assignment (Γ) results in a set of governing equations describing expected behavior. If this set of equations predicts a value that is inconsistent with the observations, the assumption represented by Γ must be wrong. That is, the mode assignment Γ is not a diagnosis.

In general there are multiple diagnoses and computing all diagnoses can be very time consuming. However most of the times we are only interested in the *most probable* (de Kleer and Williams 1989). Using the prior probabilities of the modes we first test the most likely mode assignments for consistency. If the consistency test succeeds, the posterior probability can be computed by incorporating the number of observations that are explained by the mode assignment⁹ as is described in (de Kleer and Williams 1989).

If a highly probable diagnosis Γ contains one or more fault (or abnormal) modes, it is likely that the corresponding components are the culprit.

Example 3.2: Consider the example of the mirror unit and the command interpreter again. Assume that we observe that the mirror is *not* moving after a swath command has been given. Using only these observations, we can only assume that either (or both) the mirror unit or the command interpreter is malfunctioning. However, the command interpreter controls other components

⁹Note that the mode assignment which assigns the abnormal mode to all components yields always a consistent theory, but does not explain any observation.

as well. So, if we see that, *e.g.*, the integration time is set correctly, then it is less likely that the command interpreter is malfunctioning and only the diagnosis stating that the mirror unit is the culprit remains. \square

Implementation The implementation of the diagnostic system that is currently being under development is based on the work of de Kleer, Williams and Forbus, see (de Kleer and Forbus 1993). We make use of a best-first search algorithm in order to compute the most probable diagnoses, see (de Kleer and Williams 1989) for a detailed description.

3.2 Multiple models

The problem with contemporary diagnostic systems is twofold: (1) It is hard to construct a behavior model (SD); and (2) the computation of diagnoses is very hard.

Concerning the construction of a behavior model, one has to realize that in order to obtain non trivial diagnoses more than one aspect of system behavior must be described. For example, as Davis (Davis 1984) points out, for the detection of a solder-bridge between two pins of an IC, not only a electrical but also a geometrical model is needed. That is, one needs different *views* on system behavior. In case of space systems a lot of aspects, like electrical, mechanical, thermal, etc., play an essential role in the behavior of a system.

Concerning the computational hardness. In general, the computation of a set of most probable diagnoses is exponential in the number of components/relations in the behavior description. This means that there is no guarantee that a set of most probable diagnoses can be computed in acceptable time.

As solution for both problems *approximations* of behavior descriptions are propounded, see *e.g.* (Struss 1992; Bos 1994; Nayak 1994). There are two special types of approximations: *weak* and *strong* abstractions.

We start with weak abstractions.

Definition 3.4 A system description SD_1 is weaker than SD_0 (the more accurate description), if everything that can be derived from SD_1 can also be derived from SD_0 .

Weak abstractions can be used to construct views, *i.e.*, models describing a single (or restricted set of) aspect of behavior. Other examples of weak abstractions include qualitative reasoning schemes (de Kleer and Brown 1984; Forbus 1984) for continuous systems, and temporal abstractions (Hamscher 1991) for digital systems. Weak abstractions can be used to speed-up reasoning using the following property:

Property 3.1 *If a combination of mode assignments (a conflict set) yields an inconsistent set (see Definition 3.3) for the weaker description, then that combination will also yield an inconsistent set for the more accurate description (Bos 1994).*

In general, reasoning over an abstraction is less costly than over the more accurate description. So, we may start reasoning over the abstractions to get (relatively) fast but coarse¹⁰ diagnoses. If we like to refine the answers, we know that the conflict set found sofar need not be considered again. In this way we can prune the search space induced by the more accurate description.

Strong abstractions are defined as¹¹:

Definition 3.5 *A system description SD_1 is stronger than SD_0 (the more accurate description), if everything that can be derived from SD_1 can also be derived from SD_0 .*

Strong abstractions can be applied where the original description allows for a choice between two or more outcomes. For example, if the original model describes that either in this time instance or in the next a certain event occurs, the strong abstraction states one of the possibilities. Strong abstractions can also be used to speed-up reasoning by using the following.

Property 3.2 *If a combination of mode assignments yields an consistent set, i.e. a diagnosis (see Definition 3.3), for the stronger description, then that combination will also be a diagnosis for the more accurate description (Bos 1994).*

So, if one chooses one of the outcomes by selecting a strong abstraction and no contradictions are found, then in the more accurate description contradictions will also not be found.

In (Struss 1992; Nayak 1994; Bos 1994) *heterogenous* frameworks for multiple models are propounded. In these frameworks it is possible to have multiple abstractions of a given models and these abstractions can be stated in different languages. For example, both a qualitative model (de Kleer and Brown 1984; Forbus 1984) and a hierarchical abstraction (Hamscher 1991) can be used as an approximation of, say, a differential model. So, a modeler can select the formalism best suited for describing (an approximation of) system behavior. The result is a partial order on system descriptions, see Figure 4 for an example. In this figure, $SD_i \rightarrow SD_j$ denotes the fact that SD_i is an (either a weak or strong) abstraction of SD_j .

¹⁰Because the abstractions are weaker than the accurate descriptions we may, for example, oversee a diagnosis.

¹¹It is important to note that stronger is not equivalent to more accurate.

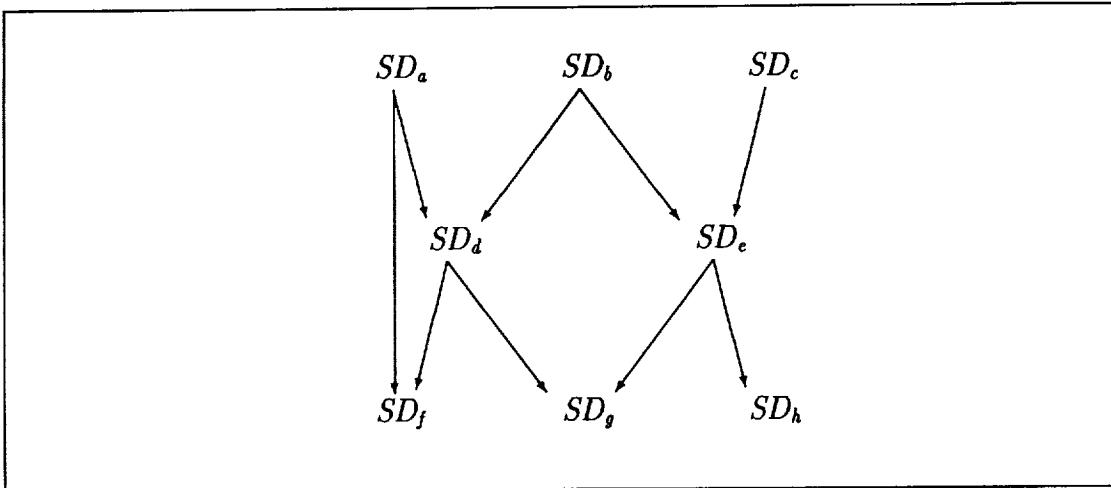


Figure 4: Abstractions of system descriptions

4 Conclusions and future work

Conclusions We have developed a monitoring system using a model-based technique. Such a system can be of great help for verifying system behavior. We have applied the monitoring system to the GOME instrument and revealed a number of discrepancies in expected and observed behavior. However, a monitoring system does not pinpoint the cause of malfunctioning; therefore, a diagnostic system should be used. A diagnostic system can be defined in a way similar to monitoring systems.

Future work We are currently developing a diagnostic system for GOME. The system will make use of abstractions in order to speed-up reasoning and to describe different aspects of system behavior.

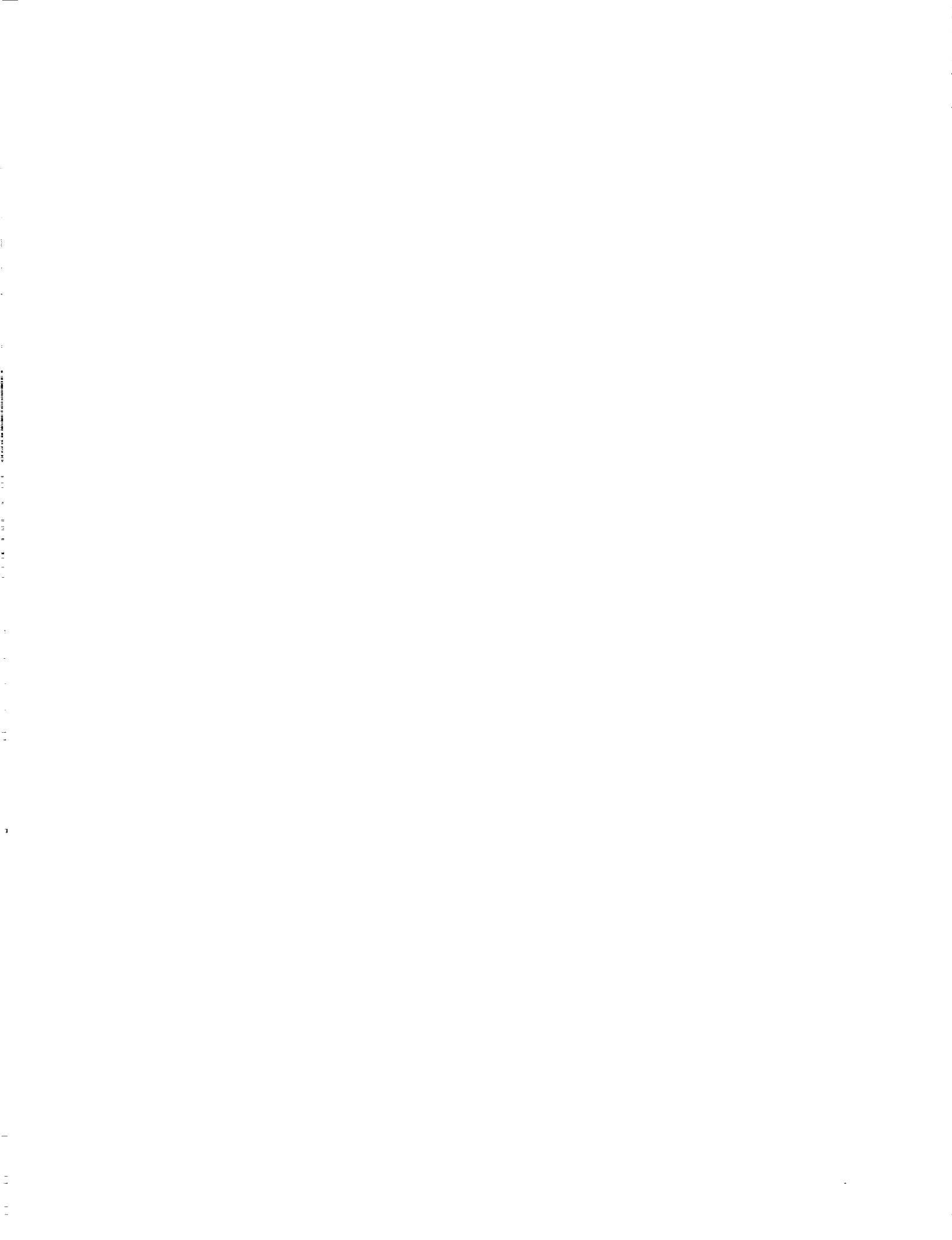
Acknowledgements

Discussions with Joost Rosier (TNO-TPD) and Cees Witteveen (Delft University of Technology) improved previous drafts considerably.

References

- Bos, A. (1994, October). Can abstractions be used to speed up diagnostic reasoning? In G. Provan (Ed.), *DX-94, The Fifth International Workshop on the Principles of Diagnosis*, New Paltz, NY, pp. 21–25.
- Davis, R. (1984). Diagnostic reasoning based on structure and behavior. *Artificial Intelligence* 24(1), 347–410.

- de Kleer, J. and J. S. Brown (1984). A qualitative physics based on confluentes. *Artificial Intelligence* 24, 7–83.
- de Kleer, J. and K. Forbus (1993). *Building Problem Solvers*. MIT Press.
- de Kleer, J. and B. C. Williams (1987). Diagnosing multiple faults. *Artificial Intelligence* 32, 97–130.
- de Kleer, J. and B. C. Williams (1989). Diagnosis with behavioral modes. In *Proceedings IJCAI-89*, pp. 1324–1330.
- ESA (1993). GOME, global ozone monitoring experiment, interim science report. Technical Report SP-1151, ESA, Noordwijk, The Netherlands.
- Forbus, K. D. (1984). Qualitative process theory. *Artificial Intelligence* 24, 85–168.
- Hamscher, W. C. (1991). Modeling digital circuits for troubleshooting. *Artificial Intelligence* 24, 223–271.
- Kernighan, B. W. and D. M. Ritchie (1978). *The C programming language*. Prentice Hall.
- Nayak, P. P. (1994, October). Diagnosis with multiple theories. In G. Provan (Ed.), *DX-94, The Fifth International Workshop on the Principles of Diagnosis*, New Paltz, NY, pp. 217–226.
- Reiter, R. (1987). A theory of diagnosis from first principles. *Artificial Intelligence* 32, 57–95.
- Struss, P. (1992). What is in SD? In W. Hamscher, J. de Kleer, and L. Console (Eds.), *Readings on Model-based Diagnosis*. Morgan Kaufmann Publishers.



1995720970

N95-27391

516-82

47266

P. 12

Linear And Nonlinear Trending And Prediction For AVHRR Time Series Data

J. Smid

Department of Mathematics

Morgan State University

Baltimore, MD 21219

e-mail smid@highwire.gsfc.nasa.gov

P. Volf, M. Slama

The Czech Academy of Sciences

Pod Vodarenskou Vezi 4

Prague 8, Czech Republic

e-mail volf@utia.cas.cz

e-mail marek@uvit.cas.cz

M. Palus

Santa Fe Institute

Old Pecos Trail

Santa Fe, NM 87505

&

The Czech Academy of Sciences

e-mail mp@santafe.edu

Abstract

The variability of AVHRR calibration coefficients in time was analysed using algorithms of linear and non-linear time series analysis. Specifically we have used the spline trend modeling, autoregressive process analysis, incremental neural network learning algorithm and redundancy functional testing. The analysis performed on available AVHRR data sets revealed that (1) the calibration data have non-linear dependencies, (2) the calibration data depend strongly on the target temperature, (3) both calibration coefficients and the temperature time series can be modeled, in the first approximation, as autonomous dynamical systems, (4) the high frequency residuals of the analysed data sets can be best modeled as an autoregressive process of the 10th degree. We have dealt with a non-linear identification prob-

lem and the problem of noise filtering (data smoothing). The system identification and filtering are significant problems for AVHRR data sets. The algorithms outlined in this study can be used for the future EOS missions. Prediction and smoothing algorithms for time series of calibration data provide a functional characterization of the data. Those algorithms can be particularly useful when calibration data are incomplete or sparse.

1. Introduction

EOS-Earth Observing System

The suite of instruments to be flown on the Earth Observing System is intended to provide a comprehensive data set of global observations of the Earth in a broad range of sensor wavelengths. Global coverage sensors will be the primary tools for data col-

lecting. Those instruments will play a major role in the global studies of our environment under the EOS project. To provide a quality data set for scientists, the raw data sets need to be calibrated. Obviously this effort requires a good calibration algorithm. We will focus in this paper on analysis of time series of calibration coefficients. This analysis can help in defining an optimum value of calibration coefficient based on the previous and current values determined from on-board calibration measurements. Also, the description of calibration data sets in terms of e.g. nonlinear dynamical systems and autoregressive processes will help in improving both the short and longterm estimation accuracy of calibration coefficients.

AVHRR Calibration Coefficients Data

Data sets from Advanced Very High Resolution Radiometer (AVHRR) sets were analysed. The AVHRR system was flown on the National Oceanographic and Atmospheric Administration (NOAA) operational meteorological satellites (NOAA-9/11). At a given time, the calibration data from one channel consists of a pair of numbers, the slope and intercept, that represent a straight line, the calibration curve. For one channel we obtain two time series, the slope as a function of time, and the intercept as a function of time. We have analysed the available data sets: year 1988, day 25; and year 1990, day 183.

The Calibration Time Series

The calibration data sets form time series. A time series $\{y(t)\}$ can be thought of as a realization of a stochastic process. A stochastic process can be described as a sequence of random variables. The best studied are linear stochastic processes. We used standard modeling techniques [Box, Jenkins 1976], based on autoregressive (AR) models to analyse the high frequency components of the data. Among our long term goals is to establish a short term, medium term and long term model for the estimate of calibration coefficients for AVHRR data. We would like to approximate calibration time series by linear

models whenever we can avoid using computationally costly non-linear models. Testing for nonlinearity is not a trivial task. Recently tests for nonlinearity based on the redundancy (entropy) functionals have been introduced [Palus, 1993] as a tool for detecting chaotic dynamical systems. We used redundancy functionals to detect nonlinear dependencies of the time series in our data sets. Knowing that we are dealing with non-linear models we would like to build a nonlinear prediction model for calibration coefficients time series as a multivariable function of external variables (the principal one is undoubtedly the target temperature). At this phase of the analysis we have used an incremental (recursive) neural network architecture to simulate the dynamics of the behavior of calibration coefficients and the target temperature for one orbit worth of data.

We have observed four main signal components in the data sets. The first component, a slow trend, corresponds to the aging of the sensors. The second trend, a nonlinear (pseudo-periodic) dependence, corresponds mainly to the effect of the day or night part of the spacecraft orbit on the sensors (the sensors are housed in the spacecraft which is affected by variable external pseudo-periodic conditions)

The third component consists of the medium frequency signal (a few minute period).

The calibration coefficients are contaminated by noise.

2. The Time Series Modeling Goals

Data Averaging

The first goal of our analysis of the AVHRR data sets was to define the current value of a calibration coefficient, i.e. the value at a given time. Since a data set of calibration coefficients is a time series contaminated with noise, this task consists of a trend estimation, by a trend modeling and a data smoothing procedure. The results of this process are time series of calibration coefficients with noise

filtered out. This process includes the estimates of confidence intervals.

Analysis of Residuals

The second goal, and a part of the data averaging procedure, is analysis of the residuals. As the series can have autocorrelated terms, it needs to be modeled by an autoregressive process.

Calibration Data Dynamics

The third goal was to investigate the modeling and predictability of the calibration coefficients in time in terms of a nonlinear dynamical system

$$dx/dt = f(x(t), t, u(t)) \quad (1)$$

where $x(t)$ represents a time series, $f()$ is an unknown non-linear function which we want to model and approximate, $u(t)$ represent external parameters (e.g. the target temperature).

Dynamics Diagnostic

A number of tests for non-linearity have been proposed [Tong 1993]. We have used a redundancy-based test for non-linearity [Palus 1993].

AVHRR Applications

Applications of modeling techniques for the AVHRR data sets are far reaching. Firstly, they will allow us, on a rigorous basis, to define the current calibration values of sensors more accurately and statistically fully qualified in terms of confidence intervals. Secondly, they will improve the long term and short term estimates of calibration coefficients through capturing the system dynamics.

3. Calibration Value Estimation Procedure

For AVHRR data sets, we made the assumption that the observed series can be described by the model

$$x(t) = q(t) + v(t) + e(t) \quad (2)$$

where t is the discrete reference time, $q(t)$ represents a long-term trend (pseudo-oscillations with period equal approximately to 102 minutes, or 1 orbit), $v(t)$ is a component describing the short-term periodicities. Finally, $e(t)$ is white noise. The first term $q(t)$ is driven by external phenomena (e.g. temperature, light intensity), and the dynamics of the sensors.

A number of methods are available for signal component modeling. We focused in the present study on the method of splines, autoregressive modeling, redundancy functionals and adaptive neural network methods.

Trend Modeling in the Time Domain (Splines)

The first task, before we can analyse noise, consists of estimating and removing the component $q(t)$ from the equation (2). A splines smoothing is the standard modeling choice [Wegman 1983]. As the character of the component $q(t)$ changes several times during one period, an algorithm is needed to capture this change in the trend. The number of knots can be controlled by the Akaike's AIC criterion [Eubank, Speckman 1990].

Dynamics Modeling in the Time Domain (Dynamical Systems)

The physical mechanisms governing the evolution of the system of calibration coefficients are not fully understood but we assumed that the data is produced by an underlying generator which is a low dimensional dynamical system.

The first component $q(t)$ is driven by external phenomena (e.g. temperature, the light intensity). This relation is complex and non-linear. In the first approximation we assumed that the 2nd order autonomous difference equations

$$dx/dt = f(x(t)) \quad (3)$$

can simulate a dynamical system that governs

the calibration coefficients. In other words we assumed that the calibration coefficients are fully described by the dynamics of the temperature evolution, which is, for simplicity, described by a *2nd order autonomous system*. The *2nd order* calibration coefficient system generates a nonlinear map

$$g : S \subseteq R^2 \rightarrow R$$

$$x(i+1) = g(x(i), x(i-1))$$

where the set S contains the training exemplars. This map can be captured by a neural network algorithm.

Dynamics Modeling in the Time Domain (Neural Networks)

We have simulated one orbit worth of the 1988 data set by the Cascor learning algorithm. Because we assumed the *2nd order* dynamics (see the previous section) we set the number of inputs for our neural net to two. The trained network creates a delay map characterizing the underlying dynamical system. Many different architectures of feedforward neural networks have been used for signal modeling and prediction [Chen, Billings, 1992]. Nonlinear system identification and modeling using neural networks has become very popular tool for modeling and identification of non-linear autonomous systems [Vemuri, 1994]. The neural network algorithms offer the flexibility of infinitely many parameters. The number of parameters is usually indirectly controlled through cross-validation. The method of cross-validation consists, in practical terms, of two phases. First a net is trained on a subset of data, secondly the trained net is tested on another data set. This process is repeated until the residual error on the test data hits its minimum. In our numerical simulation the cross validation test has been implemented using the Cascor code [Fahlman, 1992]. The follow-on tests will demonstrate the numerical agreement of the prediction and the real data. The real data for additional orbits are being prepared for neural network prediction tests.

Series Prediction by Bootstrapping

The trained net can be used for predicting the future values of the time series. We have used trained nets to predict various dynamic systems. Many authors have used different neural network architectures to predict chaotic systems. The standard test cases are logistic equation, Mackey-Glass nonlinear differential delayed equation and van der Pol equation.

The methodology of predicting, known as bootstrapping, works as follows: For an n input, m output network, a training exemplar is formed by taking $n+m$ consecutive values from the data series to be extrapolated (predicted). Starting at an arbitrary value $x(i)$, the first n values ($x(i), \dots, x(n+i-1)$) are presented to the network inputs. The target values are the next m values, thus a general training exemplar for an n input, m output network can be represented as $(x(i), \dots, x(n+i-1), \dots, x(n+m+i-1))$. Each successive exemplar is formed by starting one value beyond the previous starting value. The number of exemplars sufficient to provide the desired accuracy is to be determined by numerical experiments. After the iterative learning procedure has converged we have a map (e.g. for one output, $m=1$)

$$x(i+1) = g(x(i), \dots, x(i-n+1))$$

By bootstrapping the net into the future, this map can be iterated to give

$$x(i+2) = g(g((x(i), \dots, x(i-n+1)), x(i), \dots, x(i-n+2)))$$

The trained network can be analysed and difference equations describing the underlying dynamics can be recovered [Lowe, Webb 1994].

Noise Analysis in the Time Domain (AR-Autoregression)

The *AR* is the standard method for modeling linear dynamical systems. In the next step we have to decide about the character of the second component $v(t)$. The standard methodology recommends

analysing spectrum and periodogram, in order to decide between the trigonometric regression or autoregression. The FFT transformation, Fig.3, did not show well isolated high-frequency components of our signal. Therefore the standard methods using low-pass filters for noise removal are not directly applicable for our data sets. The periodogram revealed several isolated peaks in some regions. This feature indicated that trigonometric functions might be a good candidate for modeling of our data set. However, we have not observed this behavior of periodograms through the entire time domain. On the other hand, the partial autocorrelations showed significant dependencies of the lagged vector components for lags up to 10, less significant dependencies for the higher lags. That is why the AR approach has been preferred for statistical analysis of the series. In other words, if we denote the residuals of the first step of analysis $r(t) = x(t) - q(t)$, we have the model

$$r(t) = v(t) + e(t) = \sum_{i=1}^k b(i)r(t-i) + e(t),$$

where $b(i)$ are the parameters and k is the order of the autoregressive process AR(k).

Nonlinearity Test

The real-world data show usually some degree of nonlinearity. Our test was based on information-theoretic (redundancy) functionals. The redundancy test is based on the fact that noise, linear and nonlinear structures of a time series are represented by qualitatively different redundancy functionals.

Testing for nonlinearity of the calibration coefficients time series was performed using redundancies. Those methods were recently proposed for testing of non-linearity of dynamical systems [Palus, 1992] and are based on the general concepts of information theory.

The linear redundancy $L(X_1, \dots, X_n)$ of an n -dimensional random variable with zero mean and

covariance matrix C is defined as

$$L(X_1, \dots, X_n) = 1/2 \sum_{i=1}^n \log(c_{ii}) - 1/2 \sum_{i=1}^n \log(\sigma_i)$$

where c_{ii} are the diagonal elements and σ_i are the eigenvalues of the $n \times n$ covariance matrix C .

The n -dimensional (non-linear) redundancy is defined as

$$R(X_1, \dots, X_n) = H(X_1) + \dots + H(X_n) + H(X_1, \dots, X_n)$$

These two functionals, as the functions of the time lag τ , provide measures that differentiate linear and non-linear structures presented in the lagged versions of the component $x(t)$. The lagged version

$$(x(t), x(t+\tau), \dots, x(t+(n-1)\tau))$$

of $x(t)$ is a realization of the random variable (X_1, \dots, X_n) where n is the embedding dimension.

4. Data Characterization

Visual inspection of the data from different time periods (1988, 1990) and for different platforms revealed common features. We have searched the available data sets for features that could explain the variability of the calibration coefficients. Two features were outstanding. The long-term component $v(t)$ behaves, in the first approximation, as an autonomous dynamical system within one orbit, Fig.6.

Also PRT counts (Platinum Resistance Thermometer), located in the word 20 of the 103-word of HRPT minor frame [Kidwell, 1991] behave as an autonomous dynamical system. The target temperature is temperature of the internal target. This temperature can be calculated from the output of four PRT counts located in words 18, 19, 20 of HRPT minor frame. The conversion of PRT counts c to absolute temperature is accomplished by

$$T(K) = \sum_{j=0}^4 a_j c^j$$

- . The patch temperature (word 21) has been constant for the inspected data sets.

5. Summary of Numerical results

Our numerical results and observations are summarized in figures 1 through 17. In the following the first two digits indicate the year, and the next three digits Julian day. The slope coefficients vary significantly over one orbit. Fig.1 shows the slope coefficient series for approximately one period (88025). Fig.2 shows the PRT counts series for 88025. The FFT for the slope data (88025) did not show any significant isolated frequencies, in Fig.3. Fig.4 shows the typical relation between a slope series and PRT counts for the same time period (88025). The PRT dynamical system delay diagram (88025) shown in Fig.5 is a typical example of a map which can be learned by a neural network. The slope dynamical system delay diagram (88025) is another example of the dynamics learnable by a net (Fig.6). The PRT series (90183) shows similar features to the series (88025) even when the time difference between these series is several years (Fig.7). The same can be said about the slope values (90183), in Fig. 8. Fig. 9 is a slope vs. PRT diagram. A data smoothing procedure is illustrated in the following three figures. Fig. 10 shows a series of intercept coefficients (88025). Fig. 11 shows the smoothed intercept (88025) data. The intercept residuals (88025) showed in Fig.12 were modeled by an autoregressive process. Fig. 13 shows the best fit of the 10th order AR(10) (88025). Tests for non-linearity are captured in the remaining four figures: Fig. 14 shows slope linear redundancy statistics, dim=2 (90183). Fig. 15 shows slope linear redundancy statistics, dim=3 (90183), Fig. 16 shows slope non-linear redundancy statistics, dim=2 (9018315), Fig. 17 shows slope non-linear redundancy statistics, dim=3 (9018315). The original slope data sets were first differentiated. The results for different lags are very similar. In both

cases the difference between the linear and nonlinear measures is quite obvious (a factor of 10). That clearly indicates a need for a nonlinear description of the AVHRR data sets.

6. Conclusions

We have theorized, based on our numerical simulations, that the dynamics of the coefficients (within a time frame of days) is almost a periodic process with random input variables, due to such random effects as variable cloud cover.

A possible future application target of the proposed algorithms is short term, medium term and long term data prediction for MODIS. The MODIS instrument is scheduled to be launched in 1998. One of the important tasks in the processing of MODIS data will be to determine the most accurate value of the calibration coefficients and their corresponding uncertainties, in other words to model and identify time series generated by MODIS sensors in different wavebands. This can be accomplished by combining the predicted values furnished by a nonlinear (neural net, nonlinear AR) or linear model. A smoothing algorithm provides calibration values without noise. A prediction model will be useful especially when we dealing with the sparse or missing data. Another potential application is the detection of sudden degradation , as opposed to gradual aging, of a sensor. That will be characterized by an abrupt change in the dynamics of the sensor. To detect this change we may compare predicted values of the time series, generated by the sensor, with the new observed values. A discrepancy between the predicted and observed values flags a faulty sensor.

7. Future Work

We are in the process of using the proposed methodology for more extensive data sets. The more extensive data sets will allow us to demonstrate the

predictive power of the algorithms and to establish their error bounds. Special attention will be given to the integration algorithms for data from different calibrators (Solar Diffuser, Spectral Radiometric Calibration Assembly, Black Body, Space View), see chapter 5 in [Guenther, 1994]. Theoretical research will cover experimenting with different neural network architectures (cascade, RBF, incremental architectures). A special effort will be devoted to recovering non-linear difference equations from the trained neural network and to establishing error bounds of nonlinear predictors.

8. Acknowledgments

The GAC (The Global Area Coverage) AVHRR data were made available by M.James, Code 902, NASA/GSFC.

P.Abel, NASA/GSFC and R.Rashkin, CSC at NASA/GSFC are acknowledged for valuable comments and helpful discussions.

E.Masuoka is acknowledged for making available the computer facility of the Laboratory of Terrestrial Physics, NASA/GSFC.

J.Barker is acknowledged for making available the computer facility of the MODIS Characterization and Support Team, NASA/GSFC.

J.Smid is supported by NASA/GSFC under Grant NAG 5-2686.

References

- [S.Chen and S.A.Billings,1992] *Neural networks for nonlinear dynamics system modelling and identification* , Int.J.Control, 1992, vol. 56, no. 2, 319-346.
- [G.E.P.Box and G.M.Jenkins,1976] *Time Series Analysis, Forecasting and Control*, Holden-Day, San Francisco, 1976.
- [R.L.Eubank and P.Speckman,1990] *Curve fitting by polynomial-trigonometric regression*, Biometrika 77, 1990, pp.1-9.
- [S. E. Fahlman and C. Lebiere, 1991] *The cascade correlation learning architecture*, Research Report CMU-CS-90-100.

[B.Guenther, 1994] The MODIS Calibration Plan, Version 1.0, March 1994.

[K.Kidwell, Edt., 1991] *NOAA Polar Orbiter User's Guide*, National Oceanic and Atmospheric Administration,July 1991.

[V. Kurkova, J.Smid, 1994] *An Incremental Architecture Algorithm for Feedforward Neural Nets* , in Computer Intensive Methods in Control and Signal Processing, Preprints of the European IEEE Workshop CMP'94, Prague, 1994.

[L.Lauritson, G.Nelson, F. Porto, 1979] *Data Extraction And Calibration Of TIROS-N/NOAA Radiometers*, National Oceanic and Atmospheric Administration Technical Memorandum NESS-107,November 1979.

[D.Lowe, A.R.Webb, 1990] *Time series prediction by adaptive networks: a dynamical systems perspective*, in V. Rao. Vemuri, R.D.Rogers,eds., 1994, *Artificial Neural Networks ,Forecasting Time Series*, IEEE Computer Society Press, 1994.

[M. Paluš, 1993] *Identifying and quantifying chaos by using information-theoretic functionals*, in *Time Series Prediction: Forecasting the Future and Understanding the Past*, A.S. Weigend and N.A. Gershenfeld, (eds.), Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XV (Addison-Wesley, Reading, Mass., 1993) pp. 387-413.

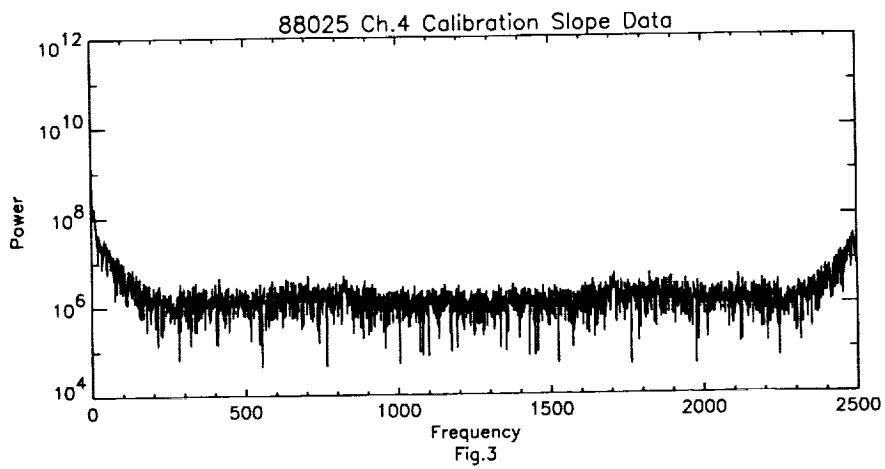
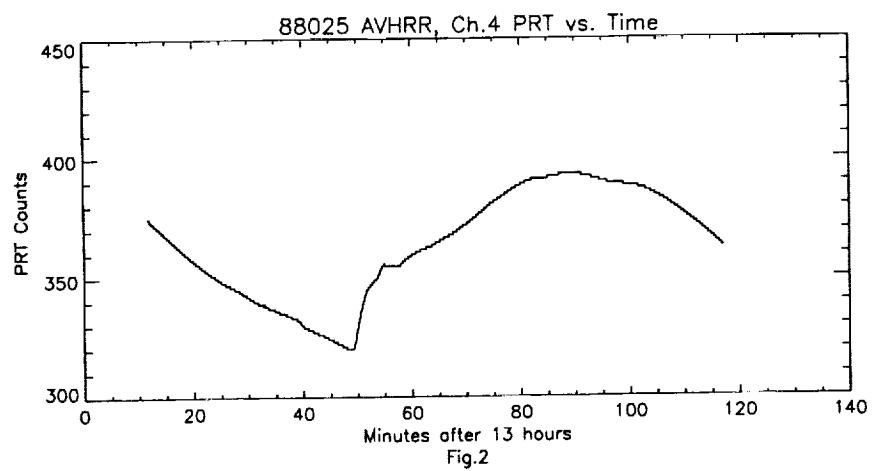
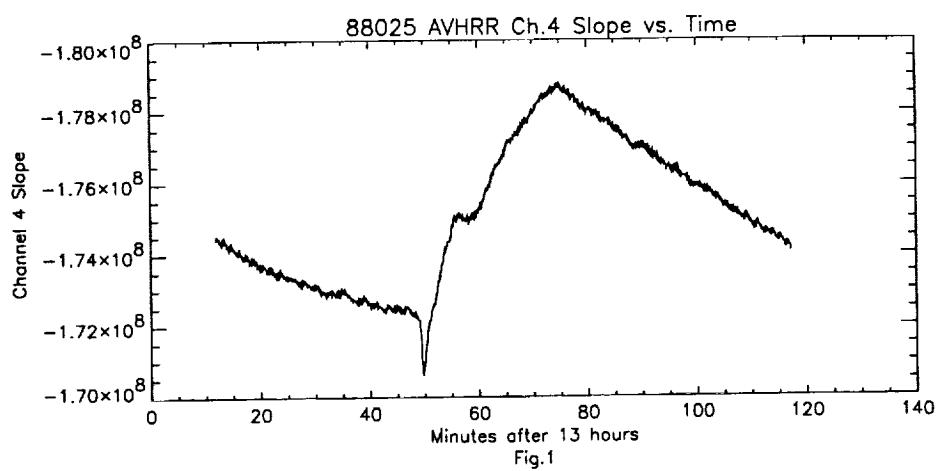
[C.R.Rao, 1965] *Linear Statistical Inference and Its Applications*, Wiley, New York, 1965.

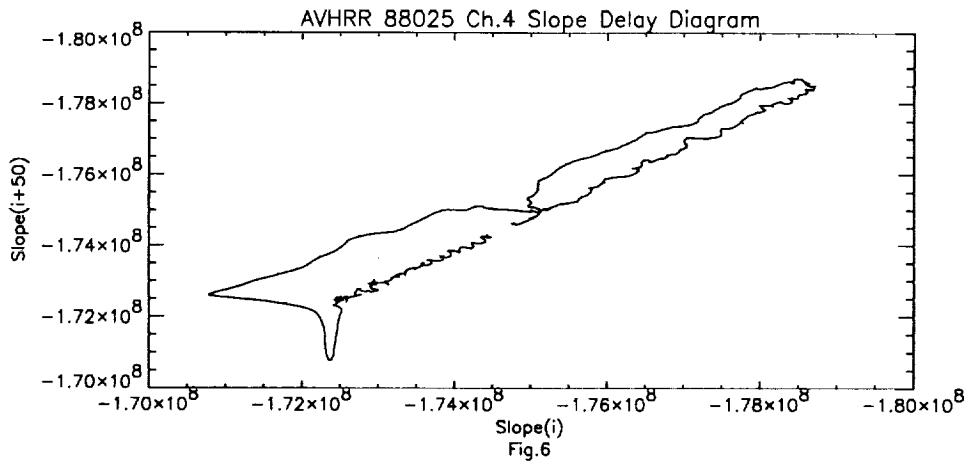
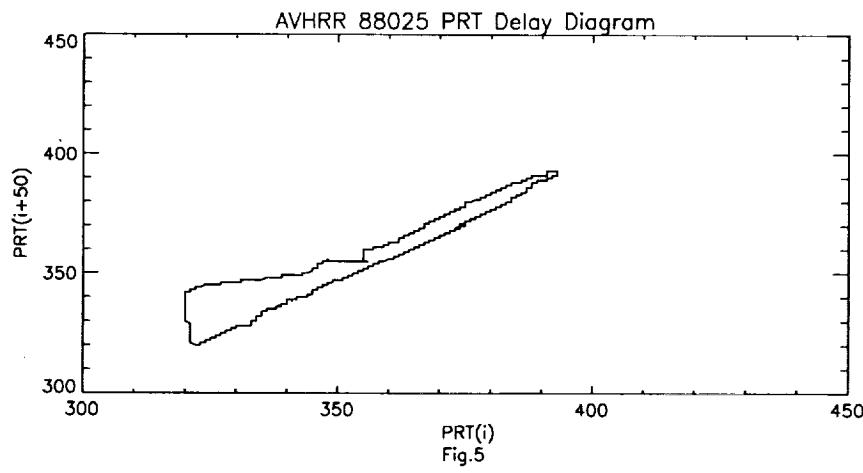
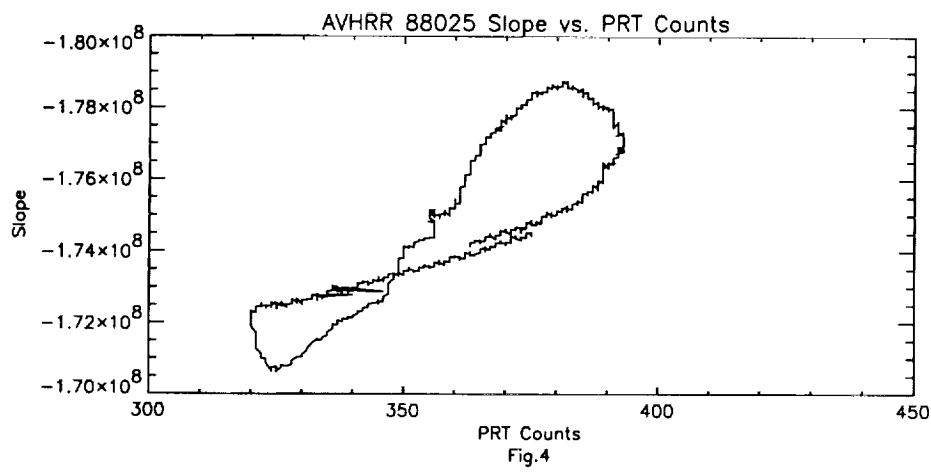
[V. Rao, Vemuri, R.D.Rogers,eds., 1994] *Artificial Neural Networks ,Forecasting Time Series*, IEEE Computer Society Press, 1994.

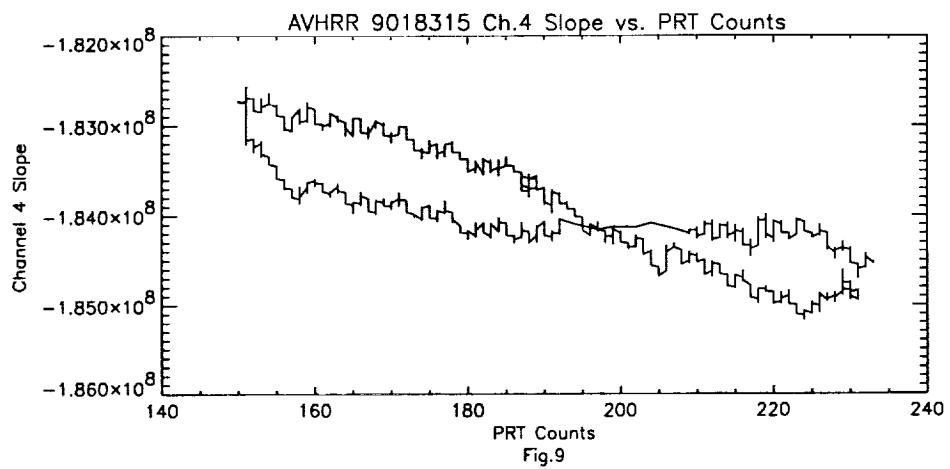
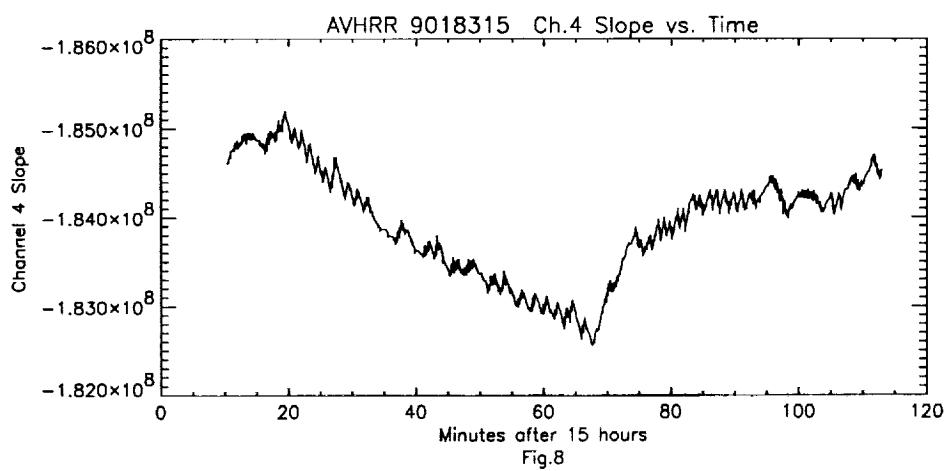
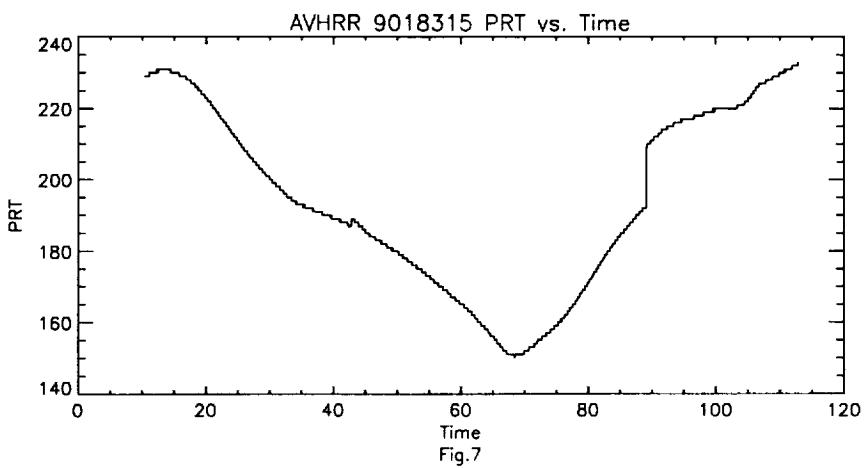
[H.Tong, 1993] *Non-linear Time Series, Dynamical System Approach* Clarendon Press Oxford, 1993.

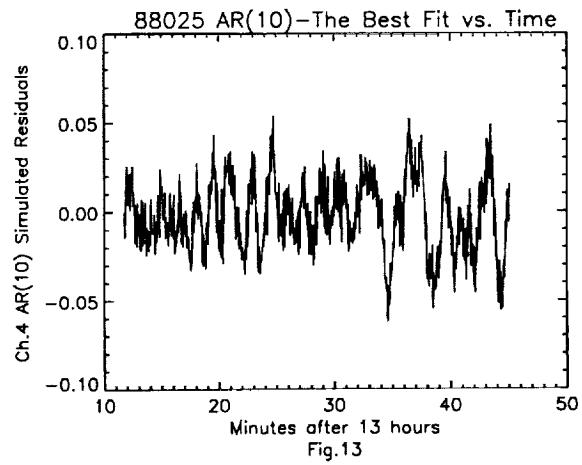
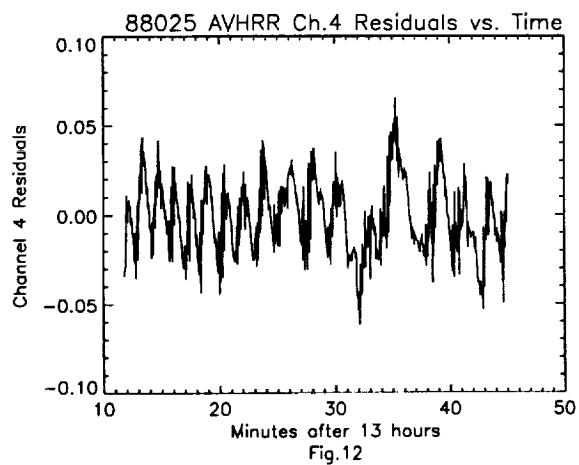
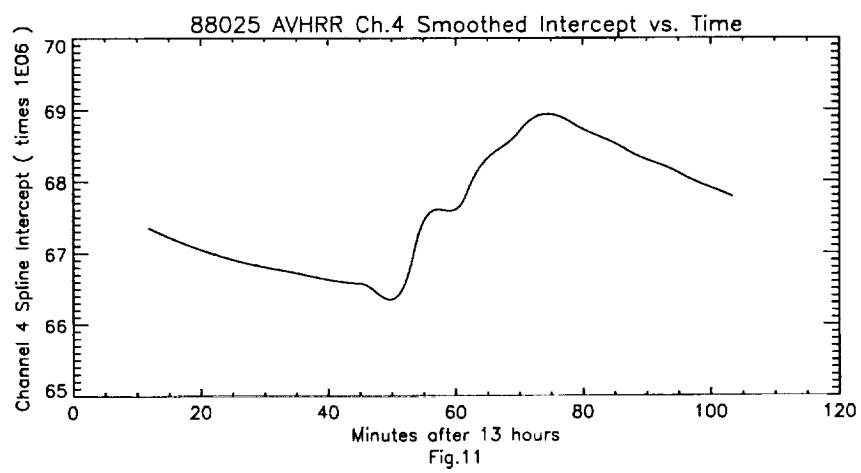
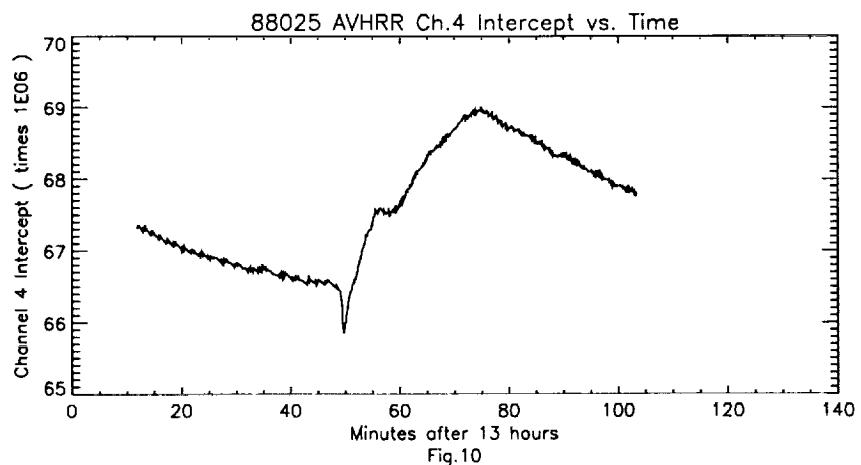
[P.Volf, 1994] *Dimension Reducing Approach in Statistical Regression Analysis*, in Computer Intensive Methods in Control and Signal Processing, Preprints of the European IEEE Workshop CMP'94, Prague, 1994.

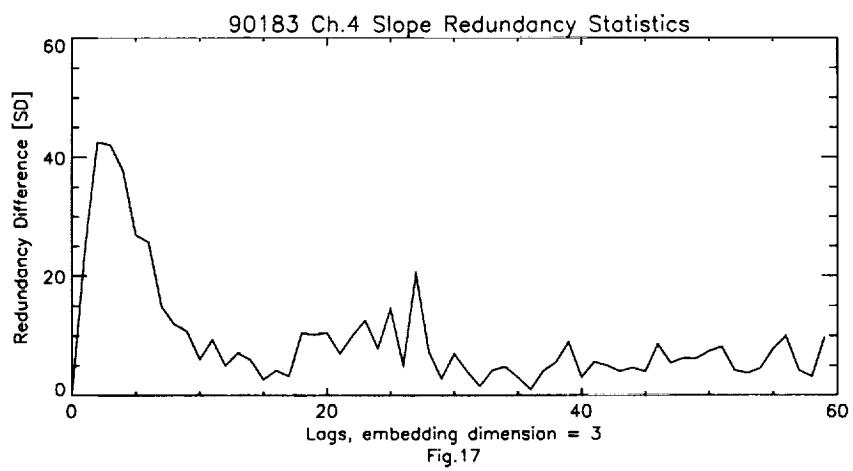
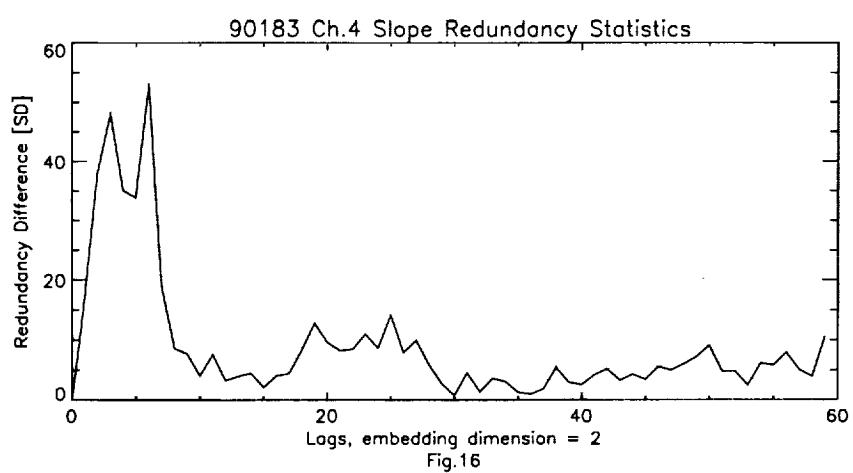
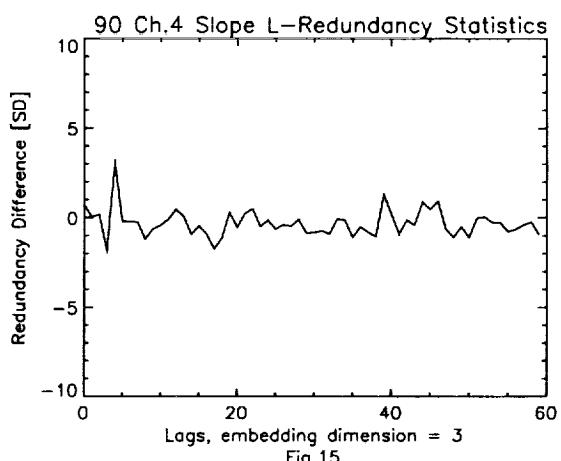
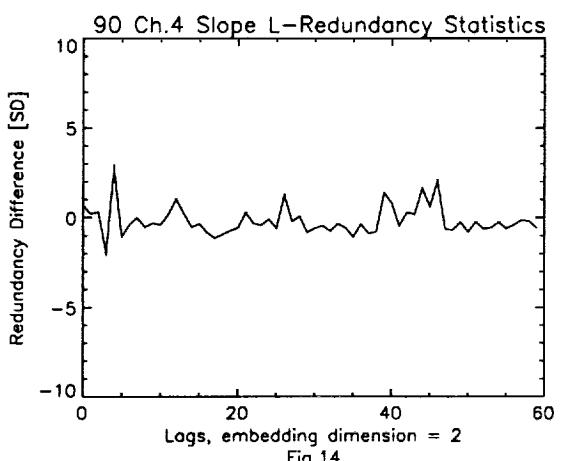
[E.J.Wegman and I.W.Wright, 1983] *Splines in statistics*, Journal of Amer. Statist. Assoc. 78, 1983, pp.351-365.











1995720971

N95-27392

A Fuzzy Logic Intelligent Diagnostic System for Spacecraft Integrated Vehicle Health Management¹

G. Gordon Wu
GeoControl Systems, Inc.
1720 NASA Road One
Houston, Texas 77058
Phone: (713) 333-2561
Fax: (713) 333-2966

5/7-63

47267

P. 7

ABSTRACT

Due to the complexity of future space missions and the large amount of data involved, greater autonomy in data processing is demanded for mission operations, training, and vehicle health management. In this paper, we develop a fuzzy logic intelligent diagnostic system to perform data reduction, data analysis, and fault diagnosis for spacecraft vehicle health management applications. The diagnostic system contains a data filter and an inference engine. The data filter is designed to intelligently select only the necessary data for analysis, while the inference engine is designed for failure detection, warning, and decision on corrective actions using fuzzy logic synthesis. Due to its adaptive nature and on-line learning ability, the diagnostic system is capable of dealing with environmental noise, uncertainties, conflict information, and sensor faults.

1. INTRODUCTION

Automated data analysis plays an important role in the success of future space

missions. The basic concept of automated data analysis is to extract data measured from existing systems, reduce them to a point where logical decisions can be deducted. Due to the complexity and the large amount of data involved, greater autonomy in data analysis and fault diagnosis is indispensable for mission operations, training, and vehicle health management.

Being a standard part of next generation spacecraft, the onboard integrated vehicle health management system will process current and historical measurement data to make failure diagnoses and corrective decisions. As an important constituent of the vehicle health management system, a diagnostic system decides which part of the measurement data to use, how to preprocess and process these data, and how to deduce the judgment and decision from the processed data. Therefore, the reliability and effectiveness of the diagnostic system are closely related to the mission success. However, the diagnostic system's performance is complicated by its working environment: the extremely large amount of the measurement data, the existence of uncertainties, and interactive vehicle operational conditions [Simpson (1994)].

In this paper, we develop a fuzzy logic intelligent diagnostic system within the frame of vehicle health management system

¹ This research is supported by NASA under contract No. NAS9-19266.

performing two major functions: (i) data reduction and information extraction; (ii) failure detection and diagnosis. These functions are performed by data filter and inference engine subsystems respectively. The data filter is designed to intelligently select only the necessary data for analysis, while the inference engine is designed to provide failure detection, warning, and corrective action decision, based on fuzzy logic synthesis and statistical analysis. Due to its adaptive nature, the diagnostic system is capable of dealing with environmental noise, uncertainties, conflict information, and sensor faults. Assisted by neural networks with learning algorithms, the system is able to conduct self-learning from previous flight data and real-time flight data.

As shown in Fig. 1, the fuzzy logic diagnostic system can be either an integrated part of the existing spacecraft control system (Fig. 1a), or an attached independent unit to assist the control system in its working process (Fig. 1b).

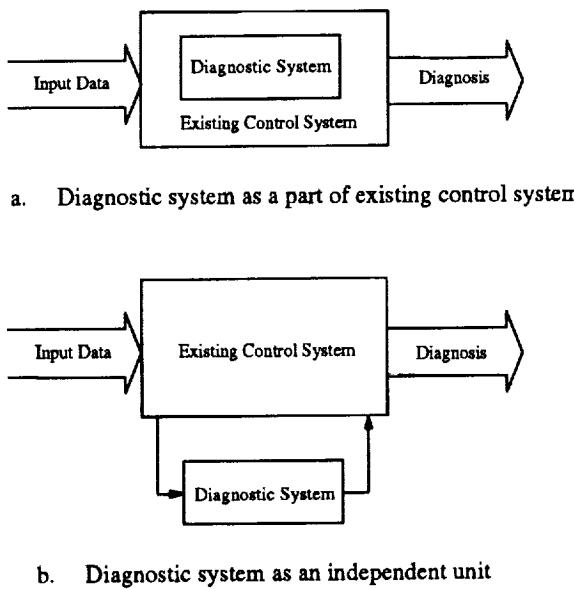


Fig. 1. Diagnostic system and control system

The diagnostic system can be easily added to and interfaced with existing control software and testbeds to accelerate the diagnostic process and to increase the precision of the diagnosis. Physically, it can be located either with ground-based control facilities or with onboard computing facilities. More specifically, it can be incorporated into the integrated vehicle health management systems for the Space Station and space shuttles.

2. DIAGNOSTIC SYSTEM STRUCTURE

Figure 2 is a schematic diagram of the general structure of the diagnostic system. Basically, this system consists of two subsystems: data filter and inference engine. The former performs data reduction and information extraction function, while the latter performs failure detection and diagnostics function.

Data Filter. The filter works at two data sampling frequencies. The important data are collected and sent to the inference engine with a high frequency. Conversely, the less important data are collected and sent to the inference engine with a low frequency. The fuzzy inference engine assigns each data source into one of the frequency groups. Data are represented by their current measurements, long-term characteristic functions, and short-term characteristic functions. Meanwhile, these data representations are also stored in a relational database. Fuzzy logic inference rules are used in the determination of the levels of importance for any given data source. Data fusion is performed by a fuzzy logic multiple-level, multiple-criteria aggregation algorithm. The weighting parameters of the generalized mean operator are determined by a neural network.

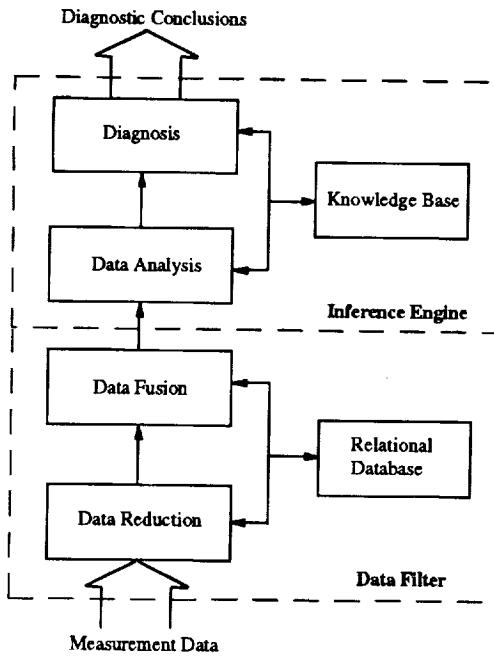


Fig. 2. System structure

Inference Engine. The inference engine has three tasks: (1) data analysis; (2) failure diagnosis; and (3) updating knowledge base. The inputs of the inference engine are the outputs of the data filter. The outputs of the inference engine are the diagnostic conclusions and corresponding corrective actions.

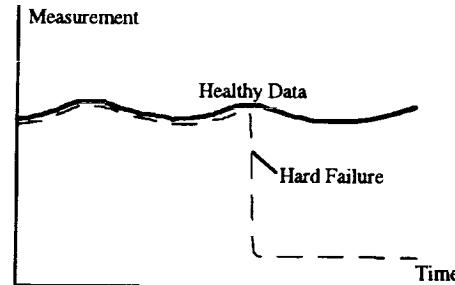
The selected fused data set comes from the data filter. Data analysis is performed as the first step to decide whether any failures are existing. Fuzzy relations are used in the data processing, assisted by statistical and fuzzy clustering methods. Diagnosis of the possible system failures is conducted by the inference engine using symptom patterns and degrees of conformity methodologies. A hierarchical clustering analysis is performed to find the data subset which causes the failure. Multiple even conflict criteria are dealt with fuzzy compatibility calculations.

The knowledge base is updated during its operation to be adaptive to deal

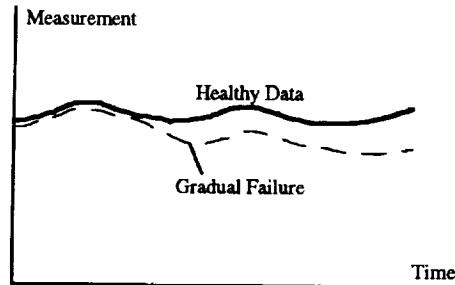
with unscheduled events, unpredictable failure, parameter changes due to system aging. A self-learning neural network is designed for the training and tuning the knowledge base during the design and development stages of the fuzzy logic system, and for adaptively updating the knowledge base in real-time operations.

3. DATA REDUCTION AND FUSION

Faulty Conditions. For the design of a diagnostic system, to have basic understanding of all the possible faulty conditions is necessary. First, there are four types of failures of a system variable with respect to time and spatial difference: hard failure (Fig. 3a); gradual failure (Fig. 3b); soft failure with full recovery (Fig. 3c); and soft failure without full recovery (Fig. 3d).

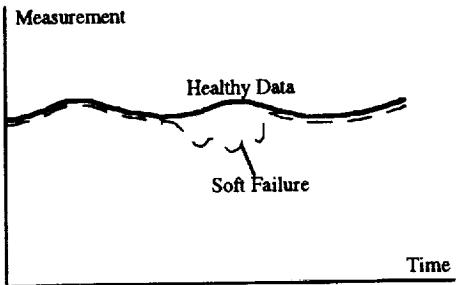


a. Hard failure

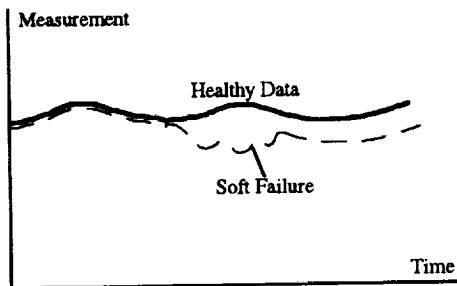


b. Gradual failure

Fig. 3. Failure types



c. Soft failure with full recovery



d. Soft failure without full recovery

Fig. 3. Failure types (continued)

Among them, all the failures are to be dealt with the diagnostic system except the hard failure which can be handled by an expert system for a quick conclusion. Next, a failure can be the result of the following: a single system failure; a single sensor failure; multiple system/sensor failures; measurement inaccuracy; and other faults including display or reading mistakes. Finally, an observed failure occurrence can be the result of a single failure or simultaneous failures.

Data Reduction. One of the important objectives of the diagnostic system is to reduce the data it uses to make failure detection and diagnosis quickly. The data filter of the fuzzy logic system is capable of performing this job intelligently [Wu (1994)]. The principle is sorting data

to different levels according to the level of importance that a specific data source has in the detection and diagnosis. The subsystem works at two data sampling frequencies. The important data are collected and sent to the inference engine at the high frequency which can be the regular frequency used for data processing. Conversely, the less important data are sampled for the inference engine at the low frequency which is a frequency set for data filtering only. If some data in the high frequency group are determined by the inference engine to be insignificant for decision making, they will be degraded to the low frequency group. Conversely, if some data in the low frequency group are determined by the inference engine as important for the decision making, they will be upgraded to the high frequency group. In this way, the data needed for decision making can be reduced considerably, while retaining all significant information.

Healthy Data and Failure

Signatures. The diagnostic system is designed for monitoring spacecraft in real time. It is impossible for the system to handle all the historical data because of the limitation of the computing facilities. However, if we only use the current measurements, we could miss a lot of important information residing in historical data. To solve this problem, we use (i) current measurements, (ii) long-term characteristic functions, and (ii) short-term characteristic functions to represent all the data. The characteristic functions vary from data source to data source, generally being fuzzy sets to store corresponding data patterns from their sources. This data representation scheme is helpful in data fusion, feature extraction, and diagnosis. For diagnostic purposes, these data representations are organized in a relational database.

In real operation, a failure in a given part of the spacecraft system comes with some abnormal features in the data measurements. It is possible for us to catch these features, i.e., its failure signature, with computational methods.

Data Fusion. Using the data representations we have, data fusion is performed by a fuzzy logic multiple-level, multiple-criteria aggregation algorithm [Loskiewicz-Buczak (1993), Barrett (1992) and Yager (1992)]. Here, the data fusion provides all possible anomaly symptoms to the inference engine instead of making conclusions by itself.

The generalized mean operator for data fusion is defined as

$$g(x_1, x_2, \dots, x_n; p; w_1, w_2, \dots, w_n) = \left\{ \sum_{i=1}^n w_i x_i^p \right\}^{1/p}, \quad (1)$$

where x_i 's are the input data with the total number of information sources as n , w_i 's are the relative importance factors to be determined for different criteria, satisfying

$$w_1 + w_2 + \dots + w_n = 1, \quad (2)$$

and p is the parameter to be determined. The generalized mean operator's values lie between the minimum and the maximum, and increase with an increase in p . By varying p between $-\infty$ and $+\infty$, the generalized mean operator can be used as union or intersection in the extreme cases. The weighting parameters w_i 's and p are determined by a neural network. See Krishnapuram (1992) for details of this procedure.

4. FUZZY INFERENCE

An inference engine is an expert system assisted by a knowledge base to

perform evaluations. The inputs of the inference engine are the outputs of the data filter. The outputs of the inference engine are the conclusions of the rules that have been fired. For our system, we use a fuzzy inference engine [Kandel (1992) and Zemankova (1989)].

Failure Diagnosis. If the inference engine determines that there is an anomaly in vehicle performance, it sends a message to the failure diagnostics subfunction. Diagnosis of the possible system failures is conducted by the inference engine using fuzzy matching of symptom patterns and degrees of conformity methodologies. Using fuzzy logic as the computational tool, a hierarchical clustering analysis is also performed to determine the data subsets causing the failure. Multiple, even conflicting criteria are dealt with by fuzzy compatibility calculations. An objective function matrix is set and adjusted in real-time operations. Neural networks are used to achieve near-optimal performance.

Besides the diagnostics function, the inference engine also presents a list of possible choices of corrective actions to vehicle manager.

Knowledge Base Updating. The knowledge base contains knowledge and human expertise. It is represented by production rules as its knowledge representation method. In the process of applications, it is updated to be adaptive for dealing with unscheduled events, unpredictable failure, parameter changes due to system aging, and to enhance the performance.

Neural Network-Fuzzy Inference Mapping. A neural network is used for tuning the fuzzy inference engine and its knowledge base during the design and

development stages of the fuzzy logic system, and for its updating in real-time operations. A mapping between the neural network and the fuzzy inference engine is needed. As a generalization of normal fuzzy logic rules, fuzzy associative memories are used to be mathematically transferred to neurons in the neural network. A modified error backpropagation algorithm is then applied to the network. Fig. 6 is a schematic of the mapping.

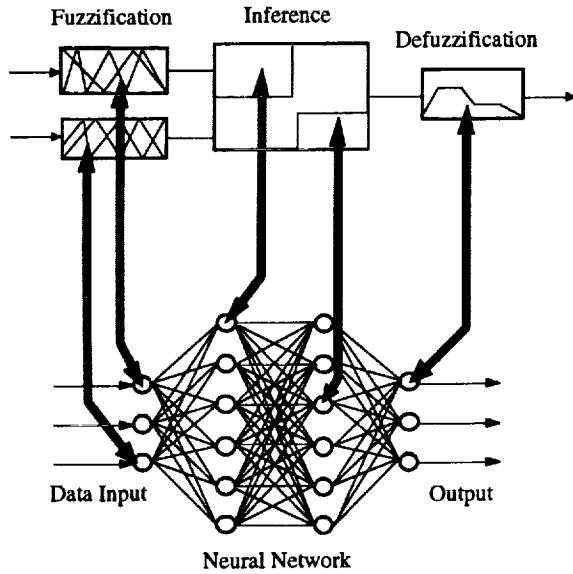


Fig. 4. Neural network-fuzzy inference mapping

5. IMPLEMENTATIONS

System Design. The project approach uses an optimization process based on performance comparison. First, the general structure is decided by study of system and performance requirements. The detailed requirements of mission operation control systems are studied to ensure that the fuzzy logic system is built to satisfy all the system requirements. Then, a comparison is conducted using different performance criteria to evaluate different structural details for the diagnostic system.

Training and Tuning. After the system is built, i.e., with its structure and algorithms decided and software developed, we proceed to the training and tuning stage in the completion of the intelligent system. First, we train the system with historical data to initiate system weights at an appropriate initial point in vector space. The knowledge base of the inference engine subsystem is then built. Meanwhile, fuzzy membership functions are tuned accordingly.

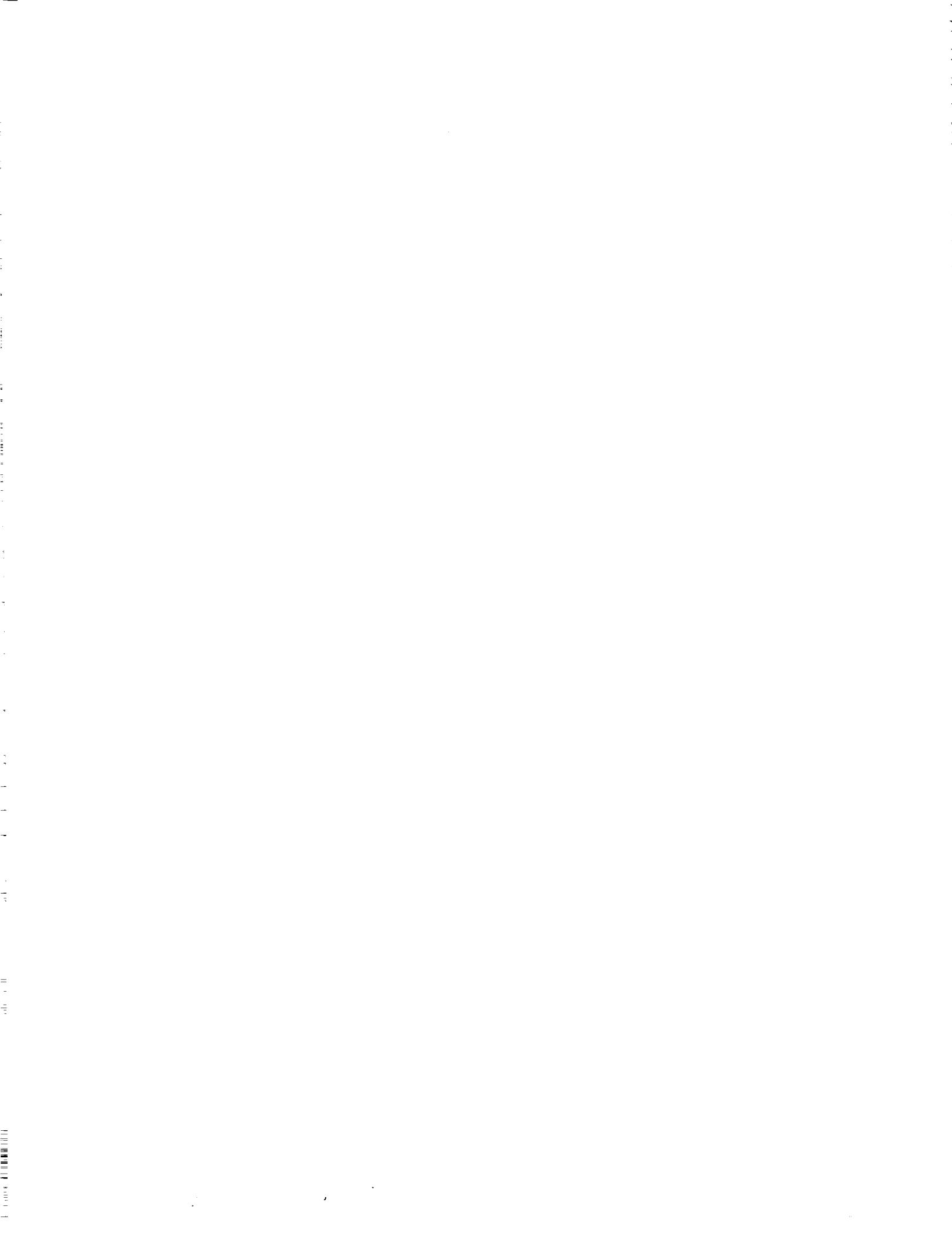
Testing and Verification. The testing and verification of the diagnostic system are conducted mainly by utilizing historical data recorded during previous flights and with existing testbeds. The use of existing testbeds greatly reduces the time and cost of system test and verification. Different performance indices are designed to test the robustness of the system and the precision of the diagnosis. Since the system can be run in parallel with existing systems, the performance of the diagnostic system is compared with that of available diagnostic techniques.

6. CONCLUSIONS

This paper discusses the principles and algorithms of a fuzzy logic diagnostic system designed for the spacecraft integrated vehicle health management. The diagnostic system contains a data filter and an inference engine. The data filter is designed to intelligently select only the necessary data for analysis, while the inference engine is designed for failure detection, warning, and decision on corrective actions using fuzzy logic synthesis. Due to its adaptive nature and on-line learning ability, the diagnostic system is capable of dealing with environmental noise, uncertainties, conflict information, and sensor faults.

REFERENCES

- [Barrett, 1992] Barrett, C. R., P. K. Pattanaik, and M. Salles, "Rationality and Aggregation of Preferences in an Ordinary Fuzzy Framework", *Fuzzy Sets and Systems*, Vol. 49, pp. 9-13, 1992.
- [Kandel, 1992] Kandel, A., Editor, *Fuzzy Expert Systems*, CRC Press, Boca Raton, Florida, 1992.
- [Krishnapuram, 1992] Krishnapuram, R., and J. Lee, "Fuzzy-Connective-Based Hierarchical Aggregation Networks for Decision Making", *Fuzzy Sets and Systems*, Vol. 46, 1992.
- [Loskiewicz-Buczak, 1993] Loskiewicz-Buczak, A., and R. E. Uhrig, "Aggregation of Evidence by Fuzzy Set Operations for Vibration Monitoring", *Proceedings of the Third International Conference on Industrial Fuzzy Control and Intelligent Systems*, pp. 204-209, Houston, Texas, 1993.
- [Simpson, 1994] Simpson, W. R., and J. W. Sheppard, *System Test and Diagnosis*, Kluwer Academic Publishers, Boston, Massachusetts, 1994.
- [Wu, 1994] Wu, G. G., "A Large-Size Data Reduction/Fusion Algorithm for Spacecraft Vehicle Health Management Systems", *Proceedings of the 1994 First International Joint Conference of NAFIPS/IFIS/NASA*, pp. 375-376, San Antonio, Texas, 1994.
- [Yager, 1992] Yager, R. R., "Higher Structure in Multi-Criteria Decision Making", *International Journal of Man-Machine Studies*, Vol. 36, pp. 553-570, 1992.
- [Zemankova, 1989] Zemankova, M., "FILIP: a Fuzzy Logic Intelligent Information System with Learning Capabilities", *Information Systems*, Vol. 14, pp. 473-486, 1989.



1995126972

N95-27393

Lessons Learned from the Introduction of Autonomous Monitoring to the *EUVE* Science Operations Center

M. Lewis, F. Girouard, F. Kronberg, P. Ringrose, A. Abedini, D. Birosnak,
T. Morgan, and R. F. Malina

5/8-63

Center for EUV Astrophysics, 2150 Kittredge St., University of California,
Berkeley, CA 94720-5030

47268
P-1

Abstract

The University of California at Berkeley's (UCB) Center for Extreme Ultraviolet Astrophysics (CEA), in conjunction with NASA's Ames Research Center (ARC), has implemented an autonomous monitoring system in the *Extreme Ultraviolet Explorer (EUVE)* science operations center (ESOC). The implementation was driven by a need to reduce operations costs and has allowed the ESOC to move from continuous, three-shift, human-tended monitoring of the science payload to a one-shift operation in which the off shifts are monitored by an autonomous anomaly detection system. This system includes Eworks, an artificial intelligence (AI) payload telemetry monitoring package based on RTworks, and Epage, an automatic paging system to notify ESOC personnel of detected anomalies.

In this age of shrinking NASA budgets, the lessons learned on the *EUVE* project are useful to other NASA missions looking for ways to reduce their operations budgets. The process of knowledge capture, from the payload controllers for implementation in an expert system is directly applicable to any mission considering a transition to autonomous monitoring in their control center. The collaboration with ARC demonstrates how a project with limited programming resources can expand the breadth of its goals without incurring the high cost of hiring additional, dedicated programmers. This dispersal of

expertise across NASA centers allows future missions to easily access experts for collaborative efforts of their own. Even the criterion used to choose an expert system has widespread impacts on the implementation, including the completion time and the final cost. In this paper we discuss, from inception to completion, the areas where our experiences in moving from three shifts to one shift may offer insights for other NASA missions.

Introduction

The *Extreme Ultraviolet Explorer (EUVE)* launched on a Delta II rocket in June of 1992. The Explorer class spacecraft carries a set of science instruments designed and built at the University of California, Berkeley (UCB). The *EUVE* mission was designed to conduct the first multi-band survey of the entire extreme ultraviolet (EUV) sky followed by spectroscopic observations of EUV sources. Mission operations are run from Goddard Space Flight Center (GSFC) while health and safety monitoring of the science payload is carried out in the *EUVE* science operations center (ESOC) at the Center for EUV Astrophysics (CEA), UCB.

Shortly after launch, it became clear that NASA's mission operations and data analysis budget faced drastic cuts. With *EUVE*'s early scientific success, CEA sought ways to dramatically lower the mission operations budget in the hope that cost reductions would allow *EUVE* to continue operating past the end of

the nominal mission. We looked at many areas of the project including the possibility of reducing staffing by introducing autonomous monitoring. Because of our lack of experience in this area, we began the process by looking for a partnership with someone possessing relevant experience. We found the NASA Code X and NASA Ames Research Center (ARC) had the knowledge and the desire to help us.

The Explorer Platform is an inherently robust spacecraft designed to last 10 years on orbit. It has both software and hardware safing conditions that can be entered with ground commanding or autonomously by the spacecraft. To date, we have never entered the hardware safehold mode but have autonomously entered the software safepointing mode twice by human error. Like the spacecraft, the science payload is very robust. The payload protects the science instruments with on-board hardware and software safety measures, such as heaters for the mirrors and control of the detector voltage level in the event a detector is being overexposed (detector doors close in the event of a serious threat). The ability of the payload and spacecraft to protect themselves from immediate threats inspired confidence for the development of a system that would detect threats of a less immediate nature, without requiring full-time human monitoring.

Selecting a Package

Lacking the time and resources to create an artificial intelligence (AI) system from scratch, we decided to evaluate off-the-shelf packages. To select an off-the-shelf package, we needed to examine what would be required of it. As a first step, we limited the scope of the problem to ensuring the safety of the science payload. Thus, a heater might be on for two days without sending anything out of limits. This situation would indicate a problem but not an immediate threat. This kind of

anomaly was not considered a priority as a human controller could notice it during the dayshift, and until a limit is exceeded it does not put the instruments at risk.

From the point of view of the *EUVE* science operations center (ESOC), we concluded that three areas require monitoring:

1. The *EUVE* science payload. This monitoring is done from telemetry that includes electrical, thermal and physical systems. Appropriate responses to changing conditions are essential to ensure the continued performance of the science instruments.
2. The communications links (Explorer Platform -> NASA -> ESOC). AI software cannot monitor the telemetry unless data are being received. If one of the communications links goes down, the software must recognize the situation and be able to summon someone to restore communications.
3. The hardware and software ground systems in the ESOC. In the event of a failure, the system must be able to summon someone to resolve the problem.

We conducted an extensive search for off-the-shelf products that would meet our needs. CEA tested products in-house for applicability, speed, and ease of use. The cost of the competing products was a factor, as was documentation and technical support. As the search progressed package stability clearly became the critical criterion. With limited manpower and a short schedule, we could not risk software deficiencies. A stable package also helps ensure the accuracy and utility of documentation. This consideration was very important, as we intended to customize the software ourselves. Several good products lacked adequate documentation. These prod-

ucts would have required us to hire the software company for implementation of the system. This would have been prohibitively expensive for our program.

Ultimately, we selected RTworks by Talarian Corporation of Mountain View, CA. RTworks displayed solid performance coupled with excellent documentation and technical support. Importantly, the generalized nature of the RTworks tools allows customizing to our needs. Moreover, the open architecture allows us to easily plug in previously existing code.

System Overview

The ESOC was formerly staffed 24 hours per day, 7 days per week by a payload controller and an engineering aide student. The ESOC is now staffed for only one 8 hour shift per day, 7 days per week. During the off shifts, the customized version of RTworks called Eworks monitors the payload. The ESOC receives telemetry, via GSFC, on a secure X.25 line. We receive real-time data, during contacts with the satellite and postpass production tape dumps. Immediately upon arrival the data are autonomously archived and decommutated using CEA software. During the dayshift, the telemetry is monitored by a controller using SOCtools (an interactive, workstation-based monitoring system developed at CEA). The data are also fed into the RTworks data acquisition module (RTdaq) and the inference engine (RTie). If Eworks detects an anomaly, requests are made to the Epage system to page an on-call payload controller. For visual monitoring of the Eworks software, the human computer-interface (RThci) module is activated.

Lessons from Implementation

We broke the implementation into two teams: one to handle the ground systems and communication issues and the other to deal with

monitoring of the science payload. Although the tasks are equally important, this paper deals primarily with the payload monitoring. The communications/ground systems group did come to a very important realization. Monitoring of communications links and local ground systems boils down to the same basic question; is the software receiving data? Or, more accurately, is the science payload being monitored? We determined that the software does not need to know the state of every link in the communications path, it only needs to know if data are being regularly supplied. So, if Eworks does not receive any telemetry for more than 6 hours the on-call controller will be paged. For more detailed information on the ground systems, see Abedini & Malina (1994).

The payload monitoring team consisted of a small group of controllers, hardware scientists, and programmers (from CEA and ARC). The team chose a small set of critical engineering monitors needed to ensure the health and safety of the science payload. The creation of an expert system was not approached by working from the hardware blueprints but rather proceeded based on expert and comprehensive knowledge of the functionality and performance of the payload. After identifying the critical areas to monitor, the team developed and tuned the method of knowledge capture.

We decided to develop an intermediate knowledge representation that would serve as a deliverable product from the domain experts to the knowledge base developers. We used informal flowcharts in a series of documents for each of the major subsystems for which we were automating the monitoring. This approach proved very useful as it cleanly separates the issues of implementation and knowledge representation from the actual knowledge itself. We had some difficulty in representing the domain knowledge in flow-

charts until we freed ourselves from the perceived need to represent the knowledge in a sequential way. On several occasions we found that we were attempting to make the knowledge representation fit into a preconceived, causal flow when it is more naturally and correctly represented by an event-driven model ("event-driven" in that nothing occurs until new data are received). The data are often received in what appears to be an asynchronous fashion because of issues of data quality, dropout, or other effects of receiving our data after the level-zero processing performed at GSFC, as well as the basic complexity of our telemetry stream.

The data-driven nature of the system in itself presents a problem since one of the very things we want to detect is a lack of current data. Ultimately, we cannot predict when we should be receiving real-time or production data since the spacecraft contact schedule often changes at the last minute. Instead we determined that it is sufficient to check whether or not data has been received within a certain number of hours. If we do not receive data from the payload for 6 hours then an alarm is raised. However, since the RTIE and in fact the whole RTworks system is driven by the reception of data, we had to create external clients that trip time-out alarms. Fortunately, the RTworks architecture provides a convenient application programming interface (API) for interfacing with custom external clients and the external clients proved easy to implement. Not only is it important that the chosen product proved flexible, but it is also important to note the recasting of the problem. While it is often essential to have an existing working system, to ensure the success of automation one must often recast what needs to be automated. Before our implementation of RTworks, we had operations personnel verify that data were received for every real-time contact, but the essential activity was the verification of pay-

load health and safety on a regular basis. No pressing need exists to verify that data are successfully received on every contact, alleviating the need to predict the contact schedule.

The nature of our telemetry format gives us problems in several areas. While existing missions rarely have the capability to change the nature of their telemetry stream, future missions should carefully examine the form in which their telemetry reaches them. The form can have profound effects upon the ease of automation. Our telemetry is level zero processed by the Packet Processing facility (PACOR) at GSFC before being sent to the ESOC. Thus, we are left in the unfortunate position of having a real-time data stream that has been stripped of all quality information. Since the data delivery format (PACOR messages) does not allow for in-band quality information but, rather, provides it at the end of each data message, we must handle reasoning on uncertain data. This format greatly complicates the implementation and can easily be avoided by providing the full data stream. In today's world of relatively cheap processing and storage resources it does not make sense to marginally compress the data delivered (stripping our quality information compresses the data by less than 5%). Other significant advantages result from keeping the original data stream, including quality, intact. For example, almost all data storage has some life expectancy beyond which the data becomes corrupted. If the full data stream is stored with the original, quality information, it can be reverified every few years.

The basic nature of our telemetry challenges us in that each frame of data does not contain a complete snapshot of all engineering channels. In our contacts with various people and various monitoring and control systems we encountered a widespread assumption that each frame of data contains a sample from all

available engineering channels. It actually takes 128 frames (over 2 minutes) of *EUVE* data to sample every engineering channel, although many are updated every one or two frames. We found that this issue, and the simple fact that the data may contain dropouts (from transmission problems), was not handled gracefully by the RTworks product.

A basic, underlying assumption is that one can reason between the last sample from every engineering channel. This assumption is inadequate because values from the current frame can only be reasoned in conjunction with the most recent expected value, which is not necessarily the most recent value received. Our interactive SOCtools package uses a shared memory segment to decouple the decommutation from display of the engineering channel values. The shared memory segment uses individual timestamps on every engineering channel to deal with this issue (and the timestamps also conveniently serve as a semaphore for multiple, asynchronous, client accesses at the individual engineering channel level).

The RTworks product does not maintain individual timestamps on the most recent values. However, because of the product's flexibility and the quality of the documentation, we were able to modify our customized RTdaq and RTie to handle this issue by supplementing the basic message types between the RTworks modules with a new message type. For gaps in the input stream, the engineering channels expected but missed are sent to the other RTworks clients in one of these new messages. In the case of the RTie receiving such a message, it sets the internal values to *unknown* for the slots corresponding to the given engineering channels. Rules do not fire when the slots they reference have *unknown* values (*unknown* is the default start-up value for all slots). In this way, all slots will either contain the most recent expected value or

unknown, and thus the integrity of the most recent value model is maintained.

Our reuse of existing code played an important role in how quickly we were able to implement our system of autonomous monitoring. Appropriate data abstractions and code modularization really paid off. The fact that much of our operations software is available through APIs has proven extremely beneficial. It enabled us to rapidly develop the previously mentioned customized RTdaq. Also, since we already had extensive limit-checking code, we did not attempt to create rules in the inference engine (RTie) to do limit checking, but rather we pass in the results of the external limit checks. This procedure has the added benefit of allowing us to easily take advantage of existing code to handle limit checking our real-time data that lacks quality information. This feat is accomplished through the use of what we call tentative limit checking. The first time a value exceeds a particular limit it is treated as only tentatively out of limits, and it is not until a second consecutive update, which also exceeds the limits, that a value is considered out of limits.

Paging

Initially we planned a combination paging and telephony system, but we were forced to scale back our efforts because of resource and schedule constraints. The current, very simple system relies on standard Unix utilities, like cron. We have postponed all efforts in the area of telephony. A key feature of our paging system is its persistence. It continues to page at regular intervals, escalating the number of people being paged after certain time-outs until an acknowledgment has been received. The system requires that the on-call personnel login to the CEA computer system and acknowledge the page(s). Ideally we would support a telephony system that

allowed the page requests to be reviewed from any phone and acknowledged by one or more button pushes. There are a number of services provided by several local and long-distance phone carriers, but to our knowledge none currently allow a customizable feedback feature (non-email based). We have found that the delivery of pages is unreliable, a fact not of common knowledge to most users. Aside from the possible human problems such as turning the pager off or forgetting to wear it, and the occurrence of dead batteries, many structures can cause shielding or interference that prevents the reception of pages. Our operations center is one such location. There is also paging service provider downtime. The low-cost solution is simply persistent pages that continue until some form of acknowledgment is made.

Another ability we did not plan for, but clearly need, is the sophisticated grouping of page requests. Our automated systems focus on the detection of problems and then bring a person into the loop. We have no automated diagnostics that can take multiple alarms and group them together into a single problem (page request). The paging system can handle an unlimited number of page requests, but the user interface is too primitive to allow convenient handling of (acknowledging and closing) multiple, simultaneous page requests.

Living with the System

As we are settling in to our new one-shift scheme, we are discovering the significance of removing humans from the control room. This move has had a profound effect on the flow of information. In the past, records were kept, but a great deal of information was exchanged face to face. During shift changes, noteworthy events could be discussed by the controllers before the ending shift departed. In our current mode of operations, controllers are separated by time and distance. As a

result, record keeping and documentation have become critical issues. A controller paged at 2 A.M. will be asleep at 8 A.M. the next morning when the dayshift arrives. In order for the members of the team to act as a cohesive unit, the records left by the paged controller must be clear, complete, and unambiguous.

We also find we are not using the system as we suspected we would. Many expert systems are designed to assist operations personnel, rather than replace them. As such, the graphical human interface is very important. In our case, the display system is secondary, since the major focus is on automating the monitoring of payload systems during unstaffed shifts. As it turns out, our rule base, so far, is not very large (< 500 rules), and over half the rules exist simply to support the human computer interface. This fact is particularly significant, as RTworks compiles the rules for the inference engine at runtime. The unnecessary rules introduce a performance penalty when developing an automated batch processing system. We are considering removing the display rules from the rule set used to process our tape dump data.

The Future

The development of our system is an ongoing process. The broadening of the rule base to include more of the engineering monitors is an obvious route for improvement. But we are working on other important areas as well. For instance, our system only reacts when something goes out of limits. It cannot predict a monitor will go out of limits based on past history and current trends. This kind of predicting is a normal part of human monitoring. We are currently working with software from NASA's Jet Propulsion Laboratory that does raise alarms based on predictions that a monitor will go out of limits. This kind of addition to our inference engine will significantly

reduce the remaining dayshift human monitoring functions, ultimately allowing a move to zero shifts. In some cases, anomalous situations may be detected early and avoided altogether.

Our current system monitors for anomalies; when an anomaly is detected, a person must be called in to deal with the problem. With the expanding capability for on-board fault detection and reaction, the next logical step is to move the autonomous monitoring software from the control center to the satellites themselves. In an ideal situation, autonomous monitoring software would have the ability to take corrective action. If the software can be taught to recognize certain types of anomalies, then it could be taught what action is necessary to deal with the situation. This applies primarily to known anomalies, but it would be an important step toward greater autonomy.

Conclusion

In the current fiscal climate, we are all going to have to find ways to reduce mission operations costs. With the development and availability of low-cost AI packages and proven mission operations software, elimination of labor intensive activities is an attainable goal. At CEA, we are proving that you can remove humans from the control room and obtain a more reliable, safer, and lower-cost operation. As our experience increases and our system matures, we become a model for other missions to follow. Likewise, our collaborative efforts across NASA centers can only help to increase the expertise available for other missions to call upon.

Acknowledgments

We would like to thank M. Montemerlo, P. Friedland, D. Korsmeyer, and D. Atkinson for their support of the development of innovative technologies for NASA missions. We

thank Dr. Guenter Riegler of NASA Headquarters and Dr. Ron Polidan and Peter Hughes of GSFC for their championing of innovation on the *EUVE* project. We would also like to thank all the members of the CEA staff who helped to make one-shift operations a reality in the *EUVE* science operations center.

This work has been supported by NASA contract NAS5-29298 and NASA Ames grant NCC2-838.

References

- Abedini, A. & Malina, R. F. 1995, Designing an Autonomous Environment for Mission Critical Operations, *Proc. SPACEOPS 1994*, "Third International Symposium on Space Mission Operations and Ground Data Systems," in press.
- Malina, R. F. 1994, Low-Cost Operations Approaches and Innovative Technology Testbedding at the *EUVE* Science Operations Center, presented at the *45th International Astronautical Congress*, IAA Symp. on Small Satellite Missions, Sess. on "Low Cost Approaches for Small Satellite Mission Operations and Data Analysis," Jerusalem, Israel, 1994 October 9-14.
- Morgan, T. & Malina R. F. 1995, Advances in Autonomous Operations for the *EUVE* Spacecraft, "Robotic Telescopes," *Proc. Astron. Soc. Pac.*, in press.
- RTworks, Talarian Corporation, 444 Castro Street, Suite 140, Mt. View, CA 94041, (415) 965-8050.



Comprehension and Retrieval of Failure Cases in Airborne Observatories*

Sergio J. Alvarado
Kenrick J. Mock

Computer Science Department
University of California, Davis
Davis, CA 95616

(916) 752-7004
alvarado@cs.ucdavis.edu
mock@cs.ucdavis.edu

5/9-12

47269

P. 15

Abstract

This paper describes research dealing with the computational problem of analyzing and repairing failures of electronic and mechanical systems of telescopes in NASA's airborne observatories, such as KAO (Kuiper Airborne Observatory) and SOFIA (Stratospheric Observatory for Infrared Astronomy). The research has resulted in the development of an experimental system that acquires knowledge of failure analysis from input text, and answers questions regarding failure detection and correction. The system's design builds upon previous work on text comprehension and question answering, including: knowledge representation for conceptual analysis of failure descriptions, strategies for mapping natural language into conceptual representations, case-based reasoning strategies for memory organization and indexing, and strategies for memory search and retrieval. These techniques have been combined into a model that accounts for: (a) how to build a knowledge base of system failures and repair procedures from descriptions that appear in telescope-operators' logbooks and FMEA (failure modes and effects analysis) manuals; and (b) how to use that knowledge base to search and retrieve answers to questions about causes and effects of failures, as well as diagnosis and repair procedures. This model has been implemented in FANSYS (Failure ANalysis SYstem), a prototype text comprehension and question answering program for failure analysis.

1. Introduction

This paper describes research dealing with the computational problem of analyzing and repairing failures of electronic and mechanical systems of telescopes in NASA's airborne observatories. To understand the reasons underlying this endeavor, consider the process of failure analysis in NASA's KAO (Kuiper Airborne Observatory), an airborne observatory for infrared astronomy (Thronson and Erickson, 1984). First, diagnosing and repairing failures of electronic and mechanical systems in KAO are time-consuming processes that can delay or cancel flights, or decrease valuable in-flight observation time. Although these processes depend on how failure-analysis information is stored and accessed, knowledge of failures in KAO is highly experiential and recorded in telescope-operator's logbooks that must be searched sequentially to determine whether given failures have occurred on previous occasions. Furthermore, even though KAO has a database of procedures for on-the-ground maintenance, this database is not designed to aid in diagnosing and repairing in-flight failures. Finally, given that NASA plans to replace KAO by building SOFIA (Stratospheric Observatory for Infrared Astronomy) (Erickson et al., 1991), then KAO can be viewed as a framework for building and experimenting with a failure-analysis system that can later be adapted for use within SOFIA.

This research has resulted in the development of an experimental system that acquires knowledge of failure analysis from input text, and answers questions regarding failure detection and correction. The

*This research was supported in part by a NASA-ASEE Faculty Fellowship to the first author, and by the NASA-Ames Research Center, Moffett Field, California, under Interchange No. NCA2-721.

model accounts for: (a) how to build a knowledge base of system failures and repair procedures from descriptions that appear in telescope-operators' logbooks and FMEA (failure modes and effects analysis) manuals; and (b) how to use that knowledge base to search and retrieve answers to questions about causes and effects of failures, as well as diagnosis and repair procedures. This model has been implemented in FANSYS (Failure ANalysis SYstem), a prototype text comprehension and question answering program (Alvarado, Braun, and Mock, 1993) initially created for the analysis of failures in the data management system (DMS) of NASA's Space Station.

FANSYS combines domain knowledge, strategies for natural language processing, and strategies for memory search and retrieval in order to understand failure descriptions and answer failure-related questions. Extending the capabilities of FANSYS into the domain of telescope-system failures has required: (1) collecting telescope design and operation logbooks and manuals to develop a model of the telescope system and a library of failure cases; (2) interviewing telescope operators and mission directors to elucidate failure modes, failure causes, failures effects, and procedures for failure detection and correction; (3) analyzing the conceptual content of the information in documents and interviews to model the knowledge structures and processes underlying the diagnosis and repair of telescope failures; and (4) implementing these structures and processes within FANSYS. As a result of these knowledge-engineering tasks, FANSYS can handle failure cases regarding a number of KAO's telescope components and systems, including the compressors, the oscillating secondary mirror (OSM), the airborne data acquisition and management system (ADAMS) and its facility interface (FI), the universal power system (UPS), the telescope inertia positioning system (TIPS), the telescope data acquisition and display system (TDADS), the tracker system, and the compensation system.

2. Comprehension and Retrieval of Telescope Failures in FANSYS

The problem of analyzing and diagnosing failures that occur in mechanical or electronic devices has been addressed by other researchers in artificial intelligence and computer science. Some of their work has been concerned with the development of truth-maintenance systems (de Kleer and Williams, 1987; Struss, 1988), expert systems (Reed and Johnson, 1990), connectionist expert systems (Liu et al., 1991), digraph-based analysis systems (Iverson and Patterson-Hine, 1990; Patterson-Hine and Iverson, 1990; Stevenson, Miller, and Austin, 1991), and case-based reasoning systems (Acorn and Walden, 1992; Hammond and Hurwitz, 1988; Redmond, 1990; Simoudis and Miller, 1990). Among these systems, there are three case-based reasoning systems that perform tasks related to those of FANSYS. First, CELIA (Redmond, 1990) learns cases regarding automobile failures, and can adapt prior solutions to solve new problems by reasoning about plans and providing justification for the new solutions. Second, CASCADE (Simoudis and Miller, 1990) retrieves failure cases regarding the VMS operating system by using a feature network and a validation network that provide access to relevant cases. Finally, Compaq's SMART system (Acorn and Walden, 1992) uses a case-based reasoning engine developed by the Inference Corporation to store and retrieve failure cases regarding Compaq's hardware and software.

Although these case-based reasoning systems deal with the problem of failure diagnosis, they do not address the problem of dynamically building the conceptual representations of failure cases from textual input, and storing those representations according to their conceptual similarities and differences. In contrast, FANSYS involves the use of techniques for parsing input failure descriptions into a conceptual network of cases that maintains the context for subsequent question answering. These techniques include: knowledge representation for conceptual analysis of failure descriptions (Alvarado et al., 1993); strategies for mapping natural language into conceptual representations (Alvarado, 1990; Riesbeck and Martin, 1986); case-based reasoning strategies for memory organization and indexing (Riesbeck and Schank, 1989); and strategies for memory search and retrieval (Dyer and Lehnert, 1982; Lehnert, 1978; Kolodner, 1984).

Comprehension and retrieval of telescope failures in FANSYS involves four major tasks: (1) applying domain-specific knowledge (i.e., telescope-system knowledge); (2) mapping input text into conceptual structures that compose the internal representation of failure cases; (3) representing and

indexing failure cases in memory by creating a knowledge base of cases; and (4) using the knowledge base of cases to answer questions regarding failure causes and effects, as well as failure detection and correction procedures. Input descriptions are segments of telescope-operator's logbooks and manuals, and contain the essential wording of the original descriptions. Here "essential" means that the original case descriptions have been edited in order to remove parts that do not fall within the following categories: failed component or system, failure mode, failure cause, failure effect, failure-detection procedure, failure-correction procedure, flight number, and flight date.

Below is an actual failure description processed by FANSYS. The description is labeled UPS-FAILURE-1 and is a fragment of a UPS failure description from a design report by NSI (1993), a contractor at NASA-Ames Research Center. The case involves a failure detected when light-emitting diodes (LEDs) on the UPS front panel indicate that the UPS is on utility power and its backup battery is disabled.

UPS-FAILURE-1

ITEM NAME: UPS (backup battery).

FAILURE MODE: Disabled UPS (backup battery) — failure during operation.

FAILURE DETECTION/VERIFICATION: The telescope operator realizes by examining the UPS LED panel (UPS-on-Utility on and Backup-Disabled on) that the UPS (backup battery) is nonoperational.

CORRECTIVE ACTION: The telescope operator switches the UPS (backup battery) on.

Text comprehension in FANSYS requires applying domain knowledge in order to build the conceptual representation of input failure descriptions. This domain-specific knowledge includes representations for each of the following classes of concepts: (1) entities, or KAO components (e.g., the UPS); (2) states, or properties of entities (e.g., the state of being nonoperational); (3) events composed of an action performed by or on a entity, the action's precondition, and the result of the action (e.g., the event of powering up the backup battery includes the power-up action, the precondition that the power be off, and the result that the power be on); and (4) procedures that consist of sequences of events involved in failure detection or correction (e.g., the read-out procedure is a detection procedure in which a failure is determined by checking indicators on a LED panel or a computer monitor).

FANSYS contains knowledge of natural language in order to map words and phrases into their appropriate conceptual structures. In FANSYS, text comprehension is performed using the case-based parsing techniques developed by Riesbeck and Martin (1986). FANSYS reads input text and questions in a left-to-right manner, one word or phrase at a time. As each word and/or phrase is read, the lexicon of FANSYS is accessed in order to identify the conceptualization underlying that word or phrase. When a lexical item is recognized, FANSYS activates an instance of the item as well as instances of patterns that encode the syntactic and semantic conditions associated with the lexical item and its underlying conceptualization. Then, FANSYS tests all pattern instances and uses those whose conditions are satisfied. The result of this process is the construction of the conceptual representation of the input.

During text comprehension, FANSYS dynamically builds the knowledge base of failure cases as a result of *remindings* (Schank, 1982) that occur when input descriptions of similar cases are read and their representations are indexed in memory using the case components. As such, text comprehension in FANSYS accounts for: (1) how cases are reorganized and how generalizations are created in memory when remindings occur; (2) how memory search and retrieval processes adapt to memory changes that occur during case indexing; and (3) how cross-contextual remindings occur when reading input failure descriptions involving different telescope components and systems. To model this reminding process, FANSYS organizes failure cases using a hierarchical-organization scheme based on the memory model developed by Kolodner (1984).

Finally, question answering in FANSYS involves using a model of memory search and retrieval developed as an extension to the question-categorization scheme of Lehnert (1978) and the retrieval heuristics of Dyer and Lehnert (1982) and Kolodner (1984). Within this model, answering a question requires analyzing the conceptual content of the question into a set of failure-case components that include

the information given and the information requested in the question. Based on this analysis, search and retrieval processes select memory indices, and traverse memory links in order to locate appropriate instances of knowledge structures to be retrieved. For example, consider the following question answered by FANSYS regarding the failures occurring in one of KAO's flights:

What is the failed component and the failure correction procedure when the flight number is 93-044?

- [1] FI (FI-TIPS LINK).
- [2] CORRECTION PROCEDURE: THE TELESCOPE OPERATOR LOGICALLY SHUTS DOWN THE TIPS. THE TELESCOPE OPERATOR LOGICALLY STARTS UP THE TIPS.
- [3] CORRECTION PROCEDURE: THE TELESCOPE OPERATOR LOGICALLY SHUTS DOWN THE FI (FI-TIPS LINK). THE TELESCOPE OPERATOR LOGICALLY STARTS UP THE FI (FI-TIPS LINK).

To answer this question, FANSYS retrieves the instantiations of the failed component and the failure-correction procedures used in flight 93-044, namely the FI-TIPS link and the rebooting procedures for the TIPS and the FI-TIPS link. Retrieving these instantiated structures requires: (1) indices and memory links from flight numbers to their associated failure cases; and (2) retrieval functions that take flight numbers as input and retrieve failed-components and failure-correction procedures from failure cases. Once the answer is found, it is converted from memory representation into adequate English by using the lexical patterns associated with each class of knowledge structure. That is, text generation in FANSYS involves using the same lexical patterns that are accessed during the text comprehension.

3. Representing Knowledge of Telescope Systems

As originally developed by Alvarado et al. (1993), the model of domain knowledge used in FANSYS encodes concepts in terms of hierarchical data structures called *Memory Organization Packets (MOPs)* (Schank, 1982; Riesbeck and Schank, 1989). Each MOP can be viewed as a *frame* (Charniak, 1977; Minsky, 1975) that consists of five major components: (1) slots, which specify the attributes and/or components of the concept encoded by the MOP; (2) fillers, which correspond to other concepts that describe values associated to the slots; (3) slot-filler links, which connect the slots of the MOP to their fillers; (4) ISA links,¹ which connect the MOP to its instances (I-MOPs) or to its specializations (M-MOPs); and (5) lexical links, which connect the MOP to associated lexical patterns used during text comprehension and text generation. This MOP-based model has been adapted to represent knowledge about five classes of concepts underlying telescope-failure descriptions, namely: entities, states, events, procedures, and cases.

3.1. Entities

An entity is defined as a component of the KAO. In FANSYS, there are two types of KAO entities: systems and messages. Systems are the hardware, software, and human components required for the operation of the KAO, such as the compressors, the telescope inertia positioning system (TIPS), and the telescope operators. Messages are the communication constructs used among or within system entities, such as network tokens, temperature-readout messages, and pressure-readout messages.

To illustrate how entities are encoded in terms of MOPs, consider figure 1, which depicts a view of a MOP hierarchy including several KAO's components. As the figure shows, M-ENTITY is connected to its two specializations, M-SYSTEM and M-MESSAGE, via arrows that denote ISA links. The MOPs that represent KAO's TIPS and compressors are shown as specializations of M-KAO-UNIT, which is a specialization of the MOP for hardware components.

¹As explained by Rich and Knight (1991), ISA links are general constructs used in knowledge representation to show that a given concept belongs to a class of concepts, such as in the phrase "a telescope operator ISA crew member."

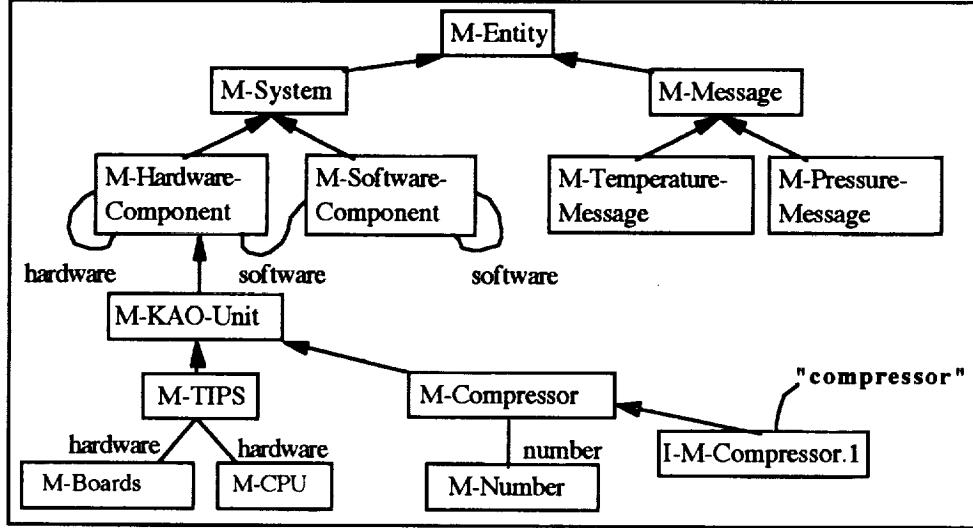


Figure 1. Partial Entity Hierarchy With ISA links (Arrows), Slot-Filler links (Single Lines), and Text Mapping.

Figure 1 also shows that system components may consist of other hardware and software subcomponents, which are connected by slot-filler links depicted as single lines. For instance, M-TIPS is composed of two hardware components, i.e., M-BOARDS and M-CPU. Similarly, M-COMPRESSOR has a number component (i.e., M-NUMBER) that uniquely identifies each compressor. Finally, figure 1 contains I-M-COMPRESSOR.1, an instance of M-COMPRESSOR that is created by FANSYS to represent the input word "compressor."

3.2. States

A state represents a relationship involving one or more KAO components. For example, consider figure 2, which shows instances of a few states. Here, the state of being on refers to the power of an individual component, whereas the state of being physically connected refers to the connection between two hardware components.

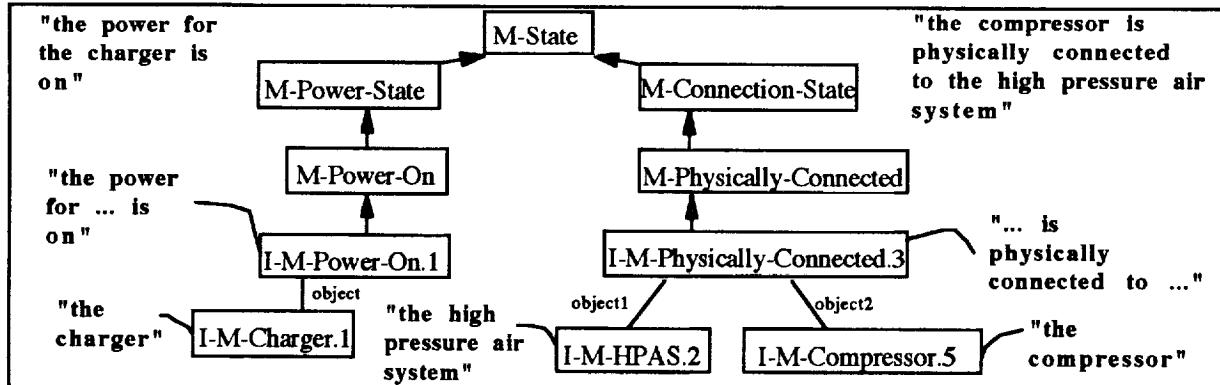


Figure 2. Instances of States.

Figure 2 indicates that states are organized in terms of the ISA and slot-filler links. Each state has one or more slots that are associated with the entities to which the state applies. For example, I-M-POWER-ON.1 has one slot for a given system, and I-M-PHYSICALLY-CONNECTED.3 has two. Figure 2 also shows how these MOPs are used to represent phrases such as "the power for the charger is

on" and "the compressor is physically connected to the high pressure air system." Within the phrases, the devices mentioned map into entity MOPs, while the textual state assertions map into the state MOPS.

3.3. Events

An event represents the occurrence of an action performed on or by an entity, the preconditions for such an occurrence, and the stage change resulting from the action. Each event and its corresponding action involve an actor, and one or more objects that are acted upon. Depending on the action encoded, events fall within one of the following types: (1) powering events, which involve turning on/off entities; (2) diagnostic events, which represent the performance of a diagnostic test; (3) connection events, which encode the establishment of physical or logical connections between different entities; and (4) transmission events, which represent the transmission and reception of messages among entities. For example, figure 3 shows how the sentence "the telescope operator powers up the charger" is represented in terms of an instance of the powering event M-POWER-UP-EVENT.

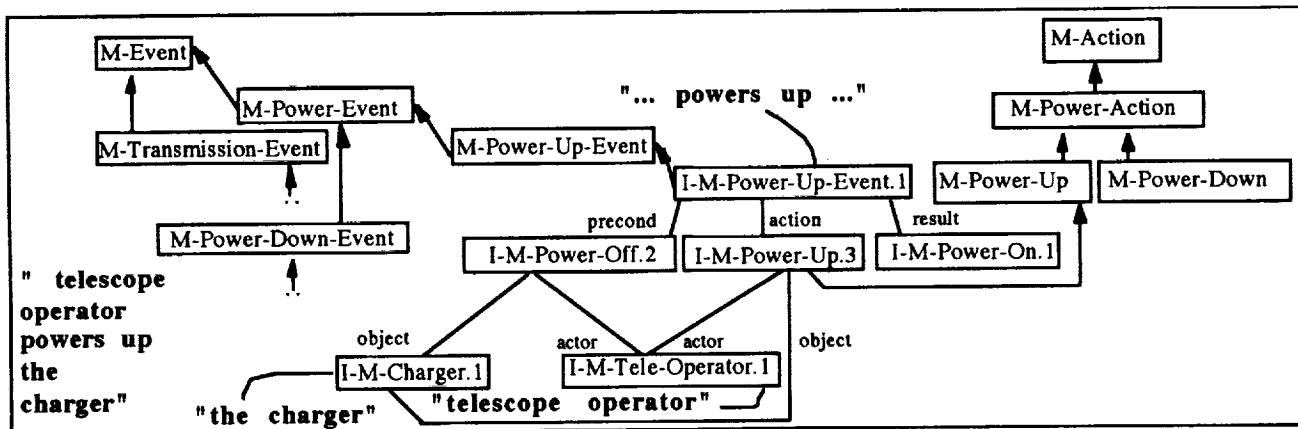


Figure 3. Instance of M-POWER-UP-EVENT.

As figure 3 indicates, I-M-POWER-UP-EVENT.1 is composed of an instance of the action M-POWER-UP, an instance of the state M-POWER-OFF that represents the precondition that the power be initially off, and an instance of the state M-POWER-ON that represents the result that the power be on. Figure 3 also shows that the phrase "powers up" maps directly into I-M-POWER-UP-EVENT.1, which makes explicit the action, precondition, and result associated with this event.

3.4. Procedures

A procedure encodes a causal chain of the steps that: (a) result in a failure mode, its cause, or its effects; or (2) can be used to detect or correct a failure. Each step in the chain contains an event or another procedure. As a result, a procedure can be viewed as a complex series of state transitions that lead from a starting set of precondition states to a final set of result states. For example, M-SOFTWARE-RESET represents a correction procedure in which telescope operators reset a system by logically shutting the failed component off, and then logically starting the failed component back on. The preconditions of this procedure include the fact that the device involved has failed and is powered on, and the result condition is that the device is operational. An instance of M-SOFTWARE-RESET is illustrated in figure 4.

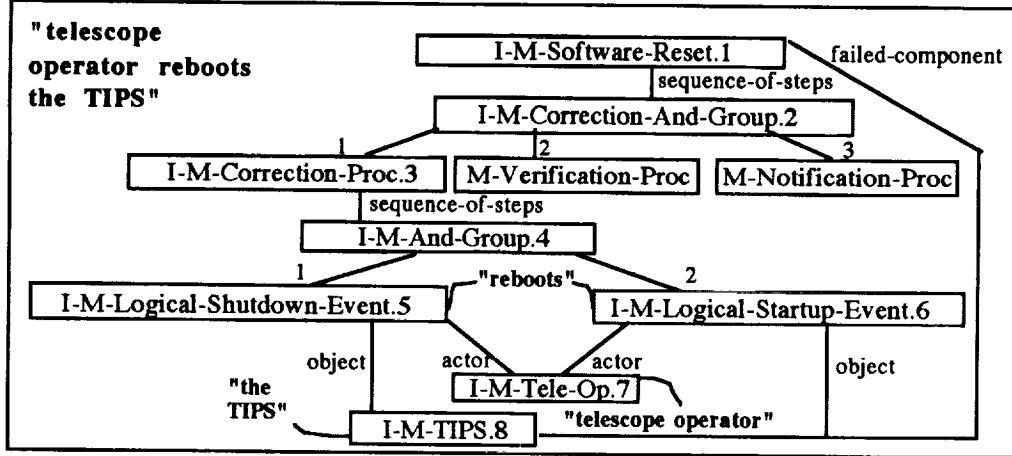


Figure 4. Instance of the procedure M-SOFTWARE-RESET.

Figure 4 shows how I-M-SOFTWARE-RESET.1 represents the sentence “the telescope operator reboots the TIPS.” In its sequence of steps, this instance MOP contains a correction procedure that makes explicit the events of logically shutting down and starting up the TIPS. The figure also shows that the representation for the telescope operator and TIPS respectively fill in the object and actor slots of I-M-LOGICAL-SHUTDOWN-EVENT.5 and I-M-LOGICAL-STARTUP-EVENT.6.

3.5. Case Representation

Cases are the top-level organizing structures that package together entities, states, events, and procedures in order to encode failure descriptions. Each case contains slots for the failed component, the failure mode, the failure cause, the failure effect, the failure-detection procedure, the failure-correction procedure, and the flight date and flight number associated with the failure. As an example, consider the text and representation of the failure description COMPRESSOR-FAILURE-1.

COMPRESSOR-FAILURE-1

ITEM NAME: Compressor 3.

FAILURE MODE: Warning Indicator— failure during operation.

FAILURE DETECTION/VERIFICATION: The telescope operator realizes from reading the ADAMS console that the pressure is low in compressor 3.

CORRECTIVE ACTION: The telescope operator switches from compressor 3 to compressor 2.

FLIGHT-DATE: 6-23-93

FLIGHT-NUMBER: 93-048

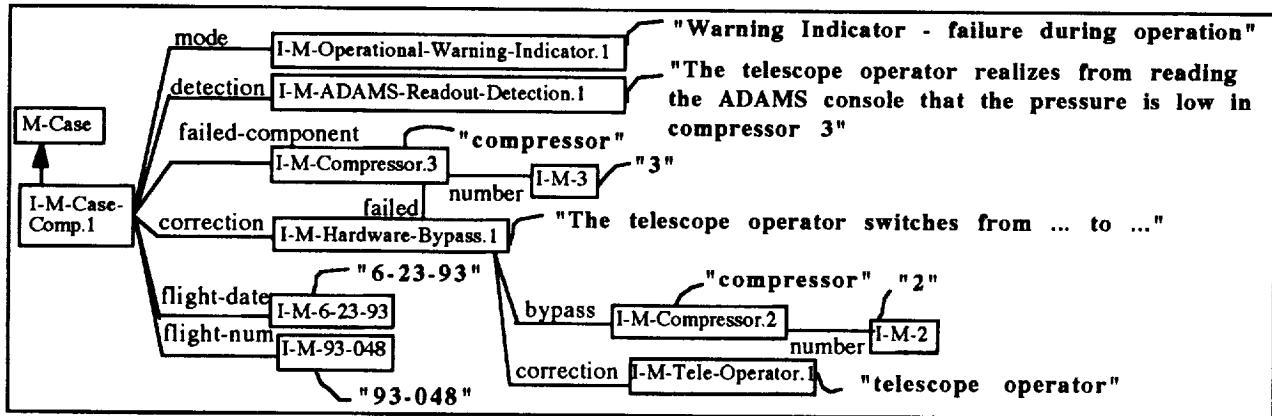


Figure 5. Case Representation of COMPRESSOR-FAILURE-1.

COMPRESSOR-FAILURE-1 indicates that a warning message regarding a compressor failure is displayed to the telescope operator, who subsequently corrects the problem by switching from compressor 3 to compressor 2. This failure description is represented in figure 5 in terms of I-M-CASE-COMP.1, a case instance indexed by the MOP M-CASE. As the figure shows, the slots of I-M-CASE-COMP.1 are filled in with instance MOPs for the given flight date, flight number, failed component, failure mode, detection procedure, and correction procedure. Figure 5 also shows a portion of the slot-filler hierarchy for the correction procedure I-M-HARDWARE-BYPASS, where I-M-COMPRESSOR.3 is the failed component, I-M-COMPRESSOR.2 is the bypass component, and I-M-TELESCOPE-OPERATOR.1 is the correction component.

4. Processing Input Text

As described by Alvarado et al. (1993), comprehension of input text and questions in FANSYS is performed using the case-based parsing techniques provided by DMAP, a Direct Memory Access Parser (Riesbeck and Martin, 1986; Riesbeck and Schank, 1989). In DMAP, parsing is a recognition process, i.e., the goal of the parser is to determine which memory structures best organize the input based upon what the parser has already been exposed to. When the parser reads an input failure description, it searches for the case representations of other descriptions previously processed in order to help it understand the input. The output of the parser is the set of memory structures referenced during the comprehension of the input, the set of new structures added to memory during parsing, and the set of expectations about what may be seen as subsequent input.

Mapping input text into its representation is accomplished through the use of MOPs called *index patterns* (Riesbeck and Martin, 1986; Riesbeck and Schank, 1989). These patterns represent stereotypical mappings of natural language onto their corresponding concepts, as well as processing knowledge that FANSYS applies to aid in constructing case representations. For example, the following pattern:

{ (correction-actor) "switches" "from" (failed) "to" (bypass) } ==> M-HARDWARE-BYPASS.

specifies that the hardware-bypass procedure can be recognized from input text if a concept representing the actor of the procedure is immediately followed by the phrase "switches from," a concept representing the failed object, the word "to," and a concept representing the bypass object. To illustrate how this pattern is used, consider a simplified, annotated trace of the inferences that FANSYS makes when reading the sentence "telescope operator switches from compressor 3 to compressor 2."

Word 1 - "telescope" - activates the pattern:

{ "telescope" "operator" } ==> M-TELESCOPE-OPERATOR.
An expectation for the word "operator" is created.

Word 2 - "operator" - fulfills the expectation from the active M-TELESCOPE-OPERATOR pattern and triggers the creation of the instance I-M-TELESCOPE-OPERATOR.1 for the concept of the telescope operator.

Word 3 - "switches" - activates the pattern:

{ (correction-actor) "switches" "from" (failed) "to" (bypass) } ==> M-HARDWARE-BYPASS.
Here, the actor is constrained to be of the type M-CREW. Given that the active concept I-M-TELESCOPE-OPERATOR.1 is an instance of M-CREW, then this instance is bound to the actor component. Expectations are created for the rest of the words in the pattern, as well as for the failed and bypass components to be of the type M-HARDWARE-COMPONENT.

The pattern { (actor) "switches" (object) "on" } ==> M-POWER-EVENT is also activated, and the actor component is bound to I-M-TELESCOPE-OPERATOR.1. Expectations are created for the rest of the words in the pattern, as well as for the object component to be of the type M-HARDWARE-COMPONENT.

Word 4 - "from" - satisfies the expectation from the M-HARDWARE-BYPASS pattern. However, this word does not match the M-POWER-EVENT pattern and, hence, this index pattern is discarded and the word "switches" is disambiguated as being part of the M-HARDWARE-BYPASS pattern.

Word 5 - "compressor" - activates the pattern { "compressor" (number) } ==> M-COMPRESSOR. An expectation is created for a number.

Word 6 - "3" - satisfies the expectation from the M-COMPRESSOR pattern and triggers the creation of the instance I-M-COMPRESSOR.3 with a filler of "3" for the number slot. This instance satisfies the expectation of the M-HARDWARE-BYPASS pattern activated by the word "switches," and I-M-COMPRESSOR.3 is bound to the failed component of the pattern.

Word 7 - "to" - satisfies the expectation from the M-HARDWARE-BYPASS pattern.

Words 8 and 9 - "compressor" "2" - are processed as words 6 and 7, i.e., the M-COMPRESSOR pattern is activated and its associated expectation is satisfied. As a result, the instance I-M-COMPRESSOR.2 is created with a filler of "2" for the number slot. Because this instance satisfies the final expectation from the M-HARDWARE-BYPASS pattern, I-M-COMPRESSOR.2 is bound to the bypass component of this pattern and the instance I-M-HARDWARE-BYPASS.1 is created.

The final result of this parsing process is the instance I-M-HARDWARE-BYPASS.1 (shown in figure 5) that has the correction component bound to I-M-TELESCOPE-OPERATOR.1, and the failed and bypass components bound to I-M-COMPRESSOR.3 and I-M-COMPRESSOR.2, respectively. In addition, the procedures, events, and states associated with I-M-HARDWARE-BYPASS.1 are also inferred.

The previous example shows the process of parsing a sentence that describes the correction procedure of a case. In addition to the index patterns used in the example, other index patterns are necessary to collect each of the case components described in the input. For example, the pattern:

{ "Corrective" "Action" ":" (correction) } ==> M-CORRECTION-CONTEXT

is used to recognize the correction procedure as the correction component of the case. Similarly, the pattern:

{ "Failed" "Item" ":" (failed-component) } ==> M-ITEM-CONTEXT

is used to parse the phrase "Failed Item: Compressor 3," and to recognize the compressor as the failed component of the case.

To build the final case representation, FANSYS must know that the input failure description may contain the failed component, the failure mode, the failure cause, the failure effect, the failure-detection procedure, the failure-correction procedure, and the flight date and flight number associated with the failure. To capture these relationships, FANSYS uses the following pattern:

{ (failed-component) (mode) (cause) (effect) (detection) (correction) (flight-number) (date) } ==> M-CASE.

This pattern is a *high level index pattern (HLIP)* (Alvarado, et al., 1993) that indicates that FANSYS should collect the representations of the case components that it finds in the input, and use them as slot fillers for an instance of a case MOP. After all the elements of the case have been collected, the final case representation is constructed, as shown in figure 5.

5. Memory Organization

The MOPs instantiated during text comprehension must be indexed so that they can be accessed during question answering. In FANSYS, memory indexing is accomplished through memory links that organize conceptual relationships among the MOPs that represent entities, states, events, procedures, and cases. As mentioned in Section 3, each MOP in FANSYS may have three types of associated links: (1)

slot-filler links, which connect the slots of the MOP to their fillers; (2) ISA links, which connect the MOP to its instances (I-MOPs) or to its specializations (M-MOPs); and (3) lexical links, which connect the MOP to associated lexical patterns used during text comprehension and text generation. However, these links do not provide a way to organize concepts based on their similarities and differences. As a result, the links are not sufficient to provide access to failure cases during question answering.

In FANSYS, the solution to this problem is to overlay an additional indexing structure on top of the memory hierarchy of ISA links. This additional indexing structure is based upon the memory representation model developed within the CYRUS system (Kolodner, 1984). Kolodner's model allows for the creation of generalizations among similar concepts, as well as the use of such generalizations as indices for the concepts. These indices form a conceptually-rich hierarchy of generalizations where many indices may point to each concept. Within the context of FANSYS, this hierarchy is termed *Generalization-Indexing (GI) memory hierarchy* (Alvarado et al., 1993). Although similar in nature to the indices used by Kolodner in CYRUS, the GI indices employed in FANSYS allow for the attributes of a given concept to be represented hierarchically.

To understand the nature of the memory links used in FANSYS, consider the graph in figure 6, which shows the representation of the event of powering up a KAO component.

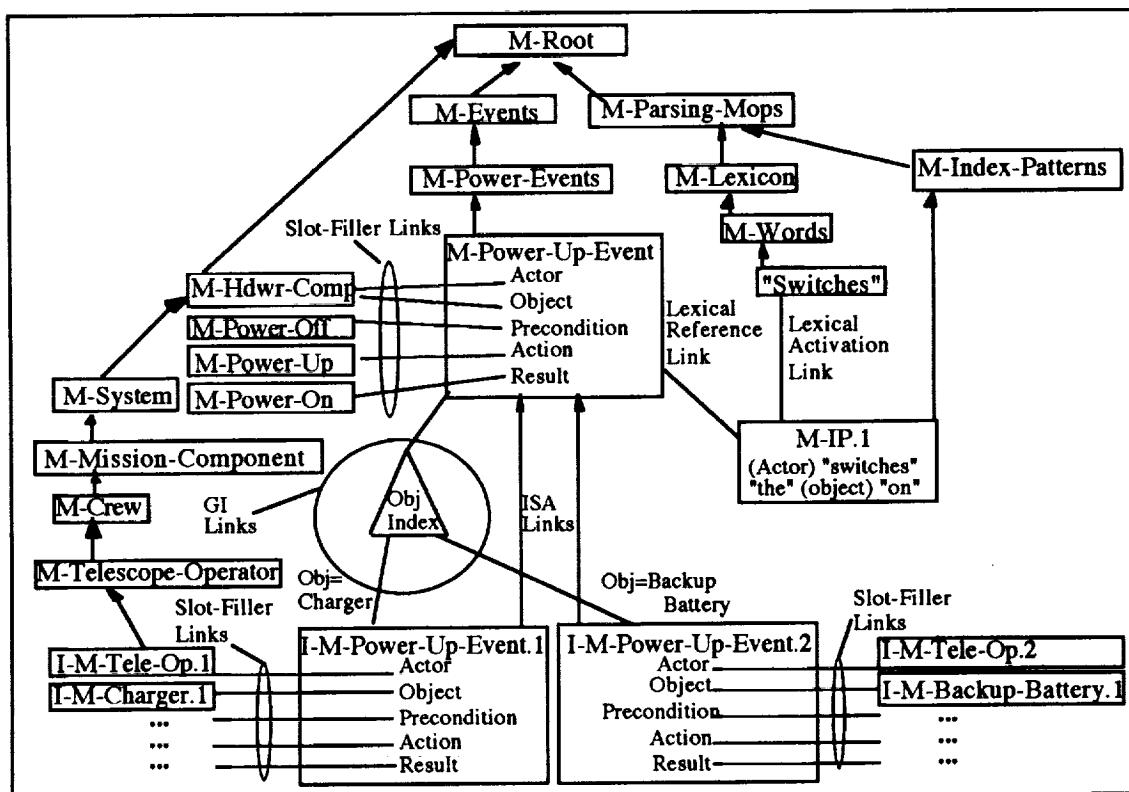


Figure 6. Types of Memory Links.

This figure contains two instances of M-POWER-UP-EVENT. One of the instances involves a telescope operator powering up a charger (I-M-POWER-UP-EVENT.1), and the other instance involves the telescope operator powering up a backup battery (I-M-POWER-UP-EVENT.2). These instances are connected to the abstraction of a general power-up-event (i.e., M-POWER-UP-EVENT) via ISA links depicted by arrowheads. These arrowheads do not indicate indexing or memory access direction, but serve only to differentiate the ISA links from other types of links. The slot-filler links indicate which actors and objects are involved in the instances of M-POWER-UP-EVENT, while the lexical links connect the pattern { (ACTOR) "SWITCHES" (OBJECT) "ON" } to M-POWER-UP-EVENT. Finally, the GI links provide a method to

use M-POWER-UP-EVENT as a generalization for indexing its two instances based upon their differences. In this example, the two instances differ in the object being powered up.

When organizing failure cases in memory, FANSYS builds a memory hierarchy in which each MOP is connected to all its specialization MOPs via GI links that encode the differences among such specializations. Furthermore, each MOP has a set of norms that contain generalized information relating to all MOPs indexed directly below it. As a result of this generalization scheme, the failure cases are located at the bottom of the hierarchy. Thus, the hierarchy organizes MOPs from the general to the specific. This generalization scheme is accomplished in FANSYS through the use of the following algorithm:

Memory-Organization Algorithm

1. Select all components of a new case as features for indexing.
2. Create a new instance MOP I1 with the components of the new case.
3. Set the current MOP CM to M-CASE, i.e., the top level MOP that indexes cases in FANSYS.
4. Set the norms of CM to the similarities between CM's norms and I1.
5. If none of the indices from CM matches I1, use the components of I1 to create indices pointing from CM to I1. Otherwise, for all indices from CM that match I1, perform one of the following:
 - 5.1. If the index points to an abstraction M1, go to step 4 with M1 as the CM.
 - 5.2. If the index points to an existing Instance MOP I2 then:
 - 5.2.1. Create a new Abstraction MOP M2.
 - 5.2.2. Incorporate the similarities between I2 and I1 as norms of M2.
 - 5.2.3. Establish the differences between I2 and I1 and use these differences to index I1 and I2 from M2.

To illustrate how the algorithm works, consider the simplified memory hierarchies shown in figure 7. Here, two failure cases involving failures of the TIPS are already indexed in memory, and a third case about a compressor failure is being integrated into memory.

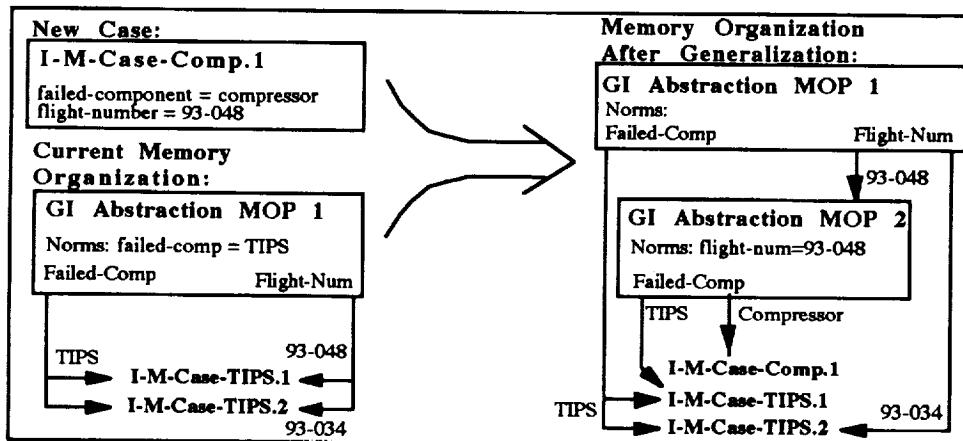


Figure 7. Example of the Application of the Memory-Organization Algorithm.

As this figure shows, I-M-CASE-COMP.1 has the same flight number as I-M-CASE-TIPS.1. When adding I-M-CASE-COMP.1 to the memory hierarchy, a match with I-M-CASE-TIPS.1 occurs along the index "Flight-Num = 93-048." This match results in the creation of the GI-Abstraction MOP2, whose norms contain the similarities between I-M-CASE-COMP.1 and I-M-CASE-TIPS.1, i.e., the flight number 93-048. In addition, the GI-Abstraction MOP2 indexes I-M-CASE-COMP.1 and I-M-CASE-TIPS.1 using their differences, i.e., the failed components. As this example illustrates, the use of the memory-organization algorithm allows FANSYS to make generalizations across failure cases and to cluster similar cases together.

6. Memory Search and Retrieval

To use the hierarchical memory organization during question answering, FANSYS applies search and retrieval strategies based on those developed by Kolodner (1984). FANSYS first parses the input question in order to represent the information given in the question and the information requested in the question in terms of the components of a failure case. Then, FANSYS compares the GI indices in the hierarchy against the question representation, and traverses the indices that match the given information in order to locate failure cases containing the requested information. This traversal process makes use of two major strategies: direct-search method and elaboration method.

The direct-search method involves performing a *depth-first search* (Rich and Knight, 1991) using the indices that match the information given in input questions. That is, FANSYS exhaustively traverses paths in its hierarchical memory that only contain each of the matching indices. As a result, only relevant MOPs within the memory hierarchy are visited. For example, consider the following question answering session:

Question: What is the failure correction procedure for the compressor when the flight number is 93-048?
Given Data: (FAILED-COMPONENT M-COMPRESSOR) (FLIGHT-NUM M-93-048)
Requested Data: (M-FAILURE-CORRECTION)
Results of Direct Search: (I-M-CASE-COMP.1)
Answer: THE TELESCOPE OPERATOR SWITCHES FROM COMPRESSOR 3 TO COMPRESSOR 2.

In this question, FANSYS is asked for the failure correction procedure used in a given flight number to fix a given component. Using the final memory organization in figure 7, FANSYS traverses the index "Flight-Num = 93-048" to reach the GI-Abstraction MOP2, and then traverses the index "FAILED-COMPONENT = Compressor" to reach the failure case I-M-CASE-COMP.1. As this example shows, the process of traversing paths composed of matching indices allows cases to be content-addressable and prevents the scanning of all possible indices.

The direct-search method only works when FANSYS is given enough information to traverse indices all the way down to the failure cases. However, FANSYS should also be capable of handling questions that do not provide enough information to traverse the memory hierarchy and reach specific cases. To solve this problem, FANSYS uses *elaboration techniques* (Kolodner, 1984) to select new indices by generating new pairs of attributes and values based on the given information and the requested information in input questions.

One elaboration method implemented in FANSYS is feature-inference elaboration, which involves relaxing the search constraints to allow the traversal of all indices that have the same attributes as the requested information in input questions. Since the indices are related to the input questions, the search is still restricted to relevant portions of memory. By allowing more indices to be traversed, there is a greater chance of retrieving cases. An example of a question answered by using the feature-inference elaboration is shown below:

Question: What is the failed component when the flight number is 93-048?
Given Data: (FLIGHT-NUM M-93-048)
Requested Data: (M-HARDWARE-COMPONENT)
Results of Direct Search: None found.
Elaborating using requested information.
Results of elaboration: (I-M-CASE-TIPS.1) (I-M-CASE-COMP.1)
Answers: [1] FI (TIPS - FI LINK) [2] COMPRESSOR (THREE)

The above question is similar to the one in the direct-search example, except that the flight number alone is not enough information to retrieve the failure cases. Using the final memory hierarchy in figure 7 and the direct-search method, traversal proceeds from the GI-Abstraction MOP1 to the GI-Abstraction MOP2 along the index "FLIGHT-NUM = 93-048." However, at this point there is no more information provided by the question and FANSYS cannot proceed any further. To retrieve the relevant cases,

FANSYS relaxes the search constraints by allowing traversal of any indices that have an attribute of FAILED-COMPONENT. As a result of the new search, the Compressor and TIPS cases can be retrieved by following the FAILED-COMPONENT indices from the GI-Abstraction MOP2.²

7. Current Status and Future Work

FANSYS has been developed at the UC Davis Artificial Intelligence Laboratory on Sun workstations using two public domain software packages: CMU Common Lisp (MacLachlan, 1992) and the GARNET User Interface Management System (Myers et al., 1992). Currently, FANSYS contains a total of fifteen cases representing failures of the KAO components listed in Section 1. These cases are built by FANSYS from input representations of case components, which are then used to create all of the indices and generalizations necessary to perform question answering. Furthermore, FANSYS contains enough lexical knowledge to parse the failure description UPS-FAILURE-1 (see Section 2), and more lexical knowledge is being implemented in FANSYS to allow it to parse other failure descriptions for the failure cases already encoded in memory. In addition, as described by Alvarado et al. (1993), the text comprehension and question answering strategies of FANSYS have previously been tested using failure descriptions involving the data management system (DMS) of NASA's Space Station.

The experience of developing an experimental failure-analysis model in FANSYS has helped elucidate representational constructs and processing strategies required for computer comprehension and retrieval of failure cases. Although FANSYS has provided an initial testbed for this model, there are limitations to FANSYS that indicate directions for future research, including the following:

(1) *Acquisition of Domain Knowledge*: FANSYS must be able to handle input text that is not directly dependent on the knowledge constructs and lexical entries initially encoded in the system. To address this knowledge-engineering issue, FANSYS must be able to dynamically augment its knowledge during text comprehension. Consequently, the model of text comprehension and question answering implemented in FANSYS must include learning strategies (Kolodner, 1993; Pazzani, 1990; Riesbeck and Schank, 1989) for acquiring and encoding knowledge constructs associated with a given domain or a given natural language.

(2) *Subjective Comprehension and Verification of Input Text*: Comprehension of input text and questions in FANSYS must also be influenced by the *ideological perspective* (Carbonell, 1981) that FANSYS may have about diagnosis and repair procedures in a given domain. That is, FANSYS must attempt to understand failure descriptions and relate their conceptual content to its own beliefs and justifications involving related and/or similar failures. If FANSYS reads descriptions that are inconsistent with its beliefs, then FANSYS must be able to generate counterarguments as it reads the input text or questions. In order to account for this process of subjective comprehension, FANSYS must have its own ideology, strategies for determining inconsistencies between its beliefs and input text, and strategies for selecting counterarguments. These counterargument strategies should be based on the argument-planning knowledge underlying the taxonomy of argument units proposed by Alvarado (1990).

(3) *Belief Inferences from Failure Cases*: FANSYS must be able to answer questions that require inferring beliefs from the cases already existing in memory. For example, consider a hypothetical question of the type: "Would FANSYS agree/disagree with advise seeker A over the use of repair procedure P when dealing with failure F?" Here, FANSYS must be able to use memories from previous failure situations to aid in understanding how a repair procedure might be used in a new failure situation. Getting FANSYS to handle hypothetical questions will require modeling the process of *belief inference* (Alvarado, 1990) in relation to *models of ideology* (Carbonell, 1981).

²If the feature-inference elaboration fails, FANSYS may use another strategy termed *alternate-context elaboration* (Kolodner, 1984), which allows FANSYS to use domain knowledge for inferring a new context for the search.

8. Conclusions

This paper has described techniques for comprehension, organization, and retrieval that are used in FANSYS to deal with failure descriptions and failure-related questions in airborne observatories. A major benefit derived from this work has been the formulation of a computational framework for: (1) integrating domain-dependent knowledge with case-based parsing strategies within text comprehension systems; (2) representing and indexing failure cases; (3) creating generalizations among these cases; and (4) using search and retrieval strategies to answer questions about failure causes and effects, as well as failure detection and correction procedures. The experience of designing and implementing this model of text comprehension and question answering has shed light on some of the basic problems any intelligent computer system must address: knowledge representation, organization, application, and retrieval. The model has helped increase understanding of representational constructs and processing strategies needed to analyze system failures. As such, FANSYS provides a computational framework for developing advice-giving and decision-making systems that may help operate and maintain NASA's aircraft and spacecraft.

References

- Acorn, T. and Walden, S. (1992). SMART: Support Management Cultivated Reasoning Technology for Compaq Customer Service. *Proceedings of IAAI-92*. Cambridge, MA: AAAI Press/MIT Press.
- Alvarado, S. J. (1990). *Understanding Editorial Text: A Computer Model of Argument Comprehension*. Boston, MA: Kluwer Academic Publishers.
- Alvarado, S. J., Braun, R. K., and Mock, K. J. (1993). *FANSYS: A Computer Model of Text Comprehension and Question Answering for Failure Analysis* (Tech. Report CSE-93-4). Computer Science Department, University of California, Davis.
- Carbonell, J. G. (1981). *Subjective Understanding: Computer Models of Belief Systems*. Ann Arbor, MI: UMI Research Press.
- Charniak, E. (1977). A framed PAINTING: The Representation of a Commonsense Knowledge Fragment, *Cognitive Science*, 1, 355-394.
- de Kleer, J. and Williams, B. (1987). Diagnosing Multiple Faults, *Artificial Intelligence*, 32, 97-130.
- Dyer, M. G. and Lehnert, W. G. (1982). Question Answering for Narrative Memory. In J. F. Le Ny and W. Kintsch (Eds.), *Language and Comprehension*. Amsterdam: North-Holland.
- Erickson, E. F., Davidson, J. A., Thorley, G., and Caroff, L. J. (1991). *SOFIA: A Stratospheric Observatory for Infrared Astronomy* (NASA Tech. Memorandum 103840). NASA-Ames Research Center, Moffett Field, CA.
- Hammond, K., and Hurwitz, N. (1988). Extracting Diagnostic Features from Explanations, *Proceedings of the Workshop on Case-Based Reasoning (DARPA)*. San Mateo, CA: Morgan-Kaufmann.
- Iverson, D. L. and Patterson-Hine, F. A. (1990). Object-Oriented Fault Tree Models Applied to System Diagnosis. *Proceedings of the SPIE Applications of Artificial Intelligence VIII Conference*. Orlando, FL.
- Kolodner, J. (1984). *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model*. Hillsdale, NJ: Lawrence Erlbaum.
- Kolodner, J. (1993). *Case-Based Reasoning*. San Mateo, CA: Morgan-Kaufmann.

- Lehnert, W. G. (1978). *The Process of Question Answering: A Computer Simulation of Cognition*. Hillsdale, NJ: Lawrence Erlbaum.
- Liu, H., Tan, A., Lim, J., and Teh, H. (1991). Practical Application of a Connectionist Expert System—The INSIDE Story, *World Congress on Expert Systems*, Orlando, FL: Pergamon Press.
- MacLachlan, R. A. (Ed.) (1992). *CMU Common Lisp User's Manual* (Tech. Report CMU-CS-92-161). School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- Minsky, M. (1975). Framework for Representing Knowledge, In P. Winston (Ed.), *Psychology of Computer Vision*. New York: McGraw-Hill.
- Myers, B. A., Giuse, D., Dannenberg, R. B., Zanden, B. V., Kosbie, D., Marchal, P., Pervin, E., Mickish, A., Landay, J. A., MacDaniel, R., and Hopkins, D. (1992). *The Garnet Reference Manual, Revised Version 2.1* (Tech. Report CMU-CS-90-117-R3). School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- NSI (1993). *UPS System Design Report #21-1117*. NSI, NASA-Ames Research Center, Moffett Field, CA.
- Patterson-Hine, F. and Iverson, D. (1990). *An Integrated Approach to System Design, Reliability, and Diagnosis*, NASA Technical Memorandum 102861. Ames Research Center, Moffett Field, CA.
- Pazzani, M. J. (1990). *Creating a Memory of Causal Relationships: An Integration of Empirical and Explanation-Based Learning Methods*. Hillsdale, NJ: Lawrence Erlbaum.
- Redmond, M. A. (1990). Distributed Cases for Case-Based Reasoning: Facilitating Use of Multiple Cases. *Proceedings of AAAI-90*. Cambridge, MA: AAAI Press/MIT Press.
- Reed, N. and Johnson, P. (1990). Generative Knowledge for Computer Troubleshooting, *European Conference on Artificial Intelligence '90*. London: Pitman.
- Rich, E. and Knight, K. (1991). *Artificial Intelligence* (Second Edition). NY: McGraw-Hill.
- Riesbeck, C. K. and Martin, C. E. (1986). Direct Memory Access Parsing. In. Kolodner, J. and Riesbeck, R. (Eds.), *Experience, Memory, and Reasoning*. Hillsdale, NJ: Lawrence Erlbaum.
- Riesbeck, C. K. and Schank, R. (1989). *Inside Case-based Reasoning*. Hillsdale, NJ: Lawrence Erlbaum.
- Schank, R. C. (1982). *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge: Cambridge University Press.
- Simoudis, E., Miller, J. (1990). Validated Retrieval in Case-Based Reasoning. *Proceedings of AAAI-90*. Cambridge, MA: AAAI Press/MIT Press.
- Stevenson, R., Miller, J., and Austin, M. (1991). Failure Environment Analysis Tool (FEAT) Development Status, *Proceedings of AIAA 1991*.
- Struss, P. (1988). Extensions to ATMS-based Diagnosis, In Gero (Ed.), *Artificial Intelligence Engineering: Diagnosis and Learning*. Amsterdam: Elsevier.
- Thronson, H. A. Jr. and Erickson, E. F. (Ed.) (1984). *Airborne Astronomy Symposium: A Symposium Commemorating the Tenth Anniversary of Operations of the Kuiper Airborne Observatory* (NASA Conference Publication 2353). NASA-Ames Research Center, Moffett Field, CA.



A Path-oriented Knowledge Representation System: Defusing the Combinatorial Explosion¹

520-63

Stefan Feyock
 Computer Science Department,
 College of William & Mary
 Williamsburg, VA 23185
 feyock@cs.wm.edu

Stamos T. Karamouzis
 National Research Council Associate
 MS 152, NASA Langley Research Center
 Hampton, VA 23681-0001
 stamos@icat.larc.nasa.gov

47270

P-10

John S. Barry
 Lockheed Engineering & Sciences
 MS 152, NASA Langley Research Center
 Hampton, VA 23681
 j.s.barry@larc.nasa.gov

Steven L. Smith
 Lockheed Engineering & Sciences
 MS 152, NASA Langley Research Center
 Hampton, VA 23681
 s.l.smith@larc.nasa.gov

Abstract

LIMAP is a programming system oriented toward efficient information manipulation over fixed finite domains, and quantification over paths and predicates. A generalization of Warshall's Algorithm to precompute paths in a sparse matrix representation of semantic nets is employed to allow questions involving paths between components to be posed and answered easily. LIMAP's ability to cache all paths between two components in a matrix cell proved to be a computational obstacle, however, when the semantic net grew to realistic size. The present paper describes a means of mitigating this combinatorial explosion to an extent that makes the use of the LIMAP representation feasible for problems of significant size. The technique we describe radically reduces the size of the search space in which LIMAP must operate; semantic nets of more than 500 nodes have been attacked successfully. Furthermore, it appears that the procedure described is applicable not only to LIMAP, but to a number of other combinatorially explosive search space problems found in AI as well.

Introduction

The set of Artificial Intelligence (AI) search/representation techniques referred to as "weak methods" generally represent problems in terms of states and operators. A typical problem description specifies one or a few start states, a test for discerning goal states, and a set of operators for generating successor states from current states. The state space is implicit, being generated by operator application as search progresses, and in many cases is potentially infinite.

¹ The research reported in this paper was supported in part by NASA Grant NCC-1-159 and NASA Contract NAS1-19000

There is a significant subset of practical AI problems, however, that involve a finite domain that is known and enumerable a priori, and unary or binary predicates over that domain; semantic net representations constitute an important instance. Well-known vector- and matrix-based representations can efficiently represent finite domains and unary/binary predicates and allow effective extraction of path information by generalized transitive closure/path matrix computations. We have developed an intelligent information tool, called LIMAP (Feyock and Karamouzis, 1991, 1992), which employs a set of abstract sparse matrix data types along with a set of operations on them as the basis for representing and manipulating finite enumerable domain (FED) problems. The present paper describes one such problem, and our experiences in attempting to apply the LIMAP system to its solution. We discuss in particular the solution we developed for overcoming the combinatorial explosion that occurred with increasing domain size, a technique that is applicable to a wide class of FED problems.

Problem Description

DRAPIYS

Semantic net representations constitute a significant subclass of FED Problems. We will begin by describing the DRAPIYS system, which employs a typical example of such a representation. DRAPIYS (Abbott, 1991) is a model-based system designed to reason about a physical system represented by its model; our group's research has centered on the modeling and diagnosis of jet engine faults. The model of the physical system is a semantic net consisting of a set of nodes, representing components such as compressors, turbines, and combustors, as well as a set of four kinds of arcs representing the relations *functionally-affects* and *physically-affects*², together with their inverses. Malfunctions are diagnosed by determining, for each component in the system, whether a possible propagation path exists linking that component to each symptomatic component. A possible propagation path is a path that satisfies the constraint that it contains no instrumented component whose sensor reading is normal.

Figure 2 and Figure 3, below, give pseudocode and LIMAP definitions of DRAPIYS's mode of operation. It is evident that DRAPIYS must determine, for each component, whether a path exists to each symptomatic component, and whether the path is in fact a possible propagation path. The original version of DRAPIYS accomplished this by means of backtracking search, a procedure that proved to be prohibitively time-consuming for large systems. Since LIMAP is a system that specializes in producing the set of paths between nodes of FEDs, it was decided to attempt to apply it to this problem.

² Component A functionally affects component B if A's malfunction can affect B's operation via a functional causal path; component A physically affects B if A's malfunction can affect B's operation via a physical path, such as fragments from A piercing B. A typical unary relation occurring in this model is *is-instrumented(A)*

The LIMAP Knowledge Representation System

LIMAP is designed to represent semantic nets, which of course are an important class of representations based on FEDs. LIMAP differs from other net-based representation systems in its emphasis on the efficient storage of large sparse nets, and on the provision of a second-order query capability oriented toward queries involving paths. Its design was motivated by the observation that queries of the form "is there a relation R such that nodes x and y are in relation xRy?" "is there a path from x to y? a path fulfilling constraint C? where can I go from x? how can I get to x?" arise frequently in AI in general, and in diagnostic problems like those addressed by DRAPHYS in particular. The following section provides a brief overview of LIMAP's query language and design; details can be found in (Feyock & Karamouzis 92).

LIMAP Overview

The LIMAP DDL/DML

The LIMAP implementation model is based on a representation that employs Boolean and symbolic vectors and adjacency matrices³ to represent unary and binary predicates, as well as an efficient transitive closure computation capability that allows Boolean or symbolic path matrices to be computed and manipulated.

As is the case for an ordinary first-order database system, LIMAP capabilities are invoked via a language interface that consists of two parts. One is the data definition language (DDL) for specifying both the data the system is to contain as well as "meta-data," that is, information about the structure and constraints that govern the data contained in the system. The other is the data manipulation language (DML), the subset of the language concerned with the specification of queries and updates on the data. We will categorize the LIMAP functions accordingly. A brief summary of the LIMAP DDL and DML follow; Feyock and Karamouzis (1991) contains a complete listing.

DDL operations: The basic DDL operations are:

```
DEFREL < name > < specification > < type > < representation >
< specification > ::= ( < number > ) | ( < number > < number > )
< type > ::= boolean | symbolic
< representation > ::= sparse | dense
```

and

```
DELREL <name>,
```

to define or delete a relation, respectively. DEFREL defines a relation by creating a new array according to the values of the parameters, and binding this array to <name>;

³ An adjacency matrix is a binary matrix representation of a graph. Entry ij is 1 iff a link joins node i and node j.

<specification> stipulates whether the array will be a vector or matrix, as well as the index range(s); <type> specifies whether the declared relation will be represented by a Boolean or symbolic array, while <representation> allows the user to choose a sparse or dense array representation.

DML operations: The major DML operations are:

STORE	relname	value	[row]	column	Store value
RETRIEVE	relname	[row]	column		Retrieve contents
TCLOSE	relname				Transitive closure
PATHS	relname	row	column		All paths
MULT	relname	relname			Multiply
TRANSPOSE	relname	relname			Transpose

STORE and RETRIEVE perform the indicated operation on the specified array position, in accordance with the array's type and representation, while MULT and TRANSPOSE typify a variety of standard matrix operations made available by LIMAP. Except in DEFREL it is transparent to the user whether the array representation is sparse or dense. This transparency extends to the other attributes of the array wherever possible.

Calculating Paths

The TCLOSE and PATHS operations form the core of LIMAP's path manipulation capability. TCLOSE computes the transitive closure of a semantic net. If G is a semantic net then the transitive closure G^* of G is a network containing an edge $\langle a, b \rangle$ if and only if G contains a path (of length 0 or greater) from a to b. TCLOSE employs Marshall's Algorithm (see, e.g., Horowitz & Sahni, 1976) for computing G^* given an adjacency matrix that represents a network G. Intuitively, the algorithm scans the matrix top to bottom, left to right. If a 1 is encountered, say in row i, column j, then row i is replaced by row i OR row j, and the scan continues from position ij.

A straightforward extension, described in (Feyock and Karamouzis, 1991), of Marshall's Algorithm to symbolic adjacency matrices produces a matrix, termed the path symbolic adjacency matrix (PSAM), whose ij entry contains the set of all paths from node i to node j. The extension, shown in Figure 1, consists of storing the paths created by appending all paths in (i,j) to all paths (j,k) into the cell (i,k). PATHS[i,j] retrieves the set of all paths from node i to node j by referencing entry ij of the resulting PSAM, thus enabling quantification over paths.

Let M be an $N \times N$ path matrix, i.e., a matrix each of whose entries contains a set of paths (represented as lists). Initially $M[i, j] = \{(i, j)\}$ i.e., the singleton set containing a one-step path from i to j, iff there is a link from node i to node j; if not, $M[i, j] = \text{NIL}$. Then after the following loop is executed, $M[i, j]$ contains the set of all paths from node i to node j.

```

for k := 1 to N do ; scan array from top down
  for i := 1 to N do ; scan array from left to right
    if (i ≠ k & M[i, k] ≠ NIL) then
      for j := 1 to N do M[i, j] := M[i, j] ∪ (M[i, k] ∥ M[k, j])

```

The \parallel operator is defined as follows:

If $p = (v_1, \dots, v_k)$ and $q = (v_k, \dots, v_r)$ then $p \parallel q = (v_1, \dots, v_k, \dots, v_r)$

Figure 1: Path Matrix Computation

Control Structures

The distinction between procedural and nonprocedural predicate calculus specifications blurs if the underlying domain is finite, since the FORALL and EXISTS quantifiers map in an obvious way to loops ranging over the domain elements. It has been our goal to give the LIMAP DML as non-procedural a character as possible. In particular, LIMAP notation is an adaptation of the (function-less) predicate calculus, with extensions to allow data retrieval in addition to data specification. Perhaps surprisingly, we have found that minimal modifications of the control macros described in (Charniak et al., 1987) were suitable for the task of expressing the required quantifications. Here is a summary of the general form of the control structure implemented by these macros:

```

(FOR ((< variable1 >: IN < set1 >)
      (< variablen >: IN < setn > ))
      [:WHEN <when-expression>]
      < FOR-keyword > < expression1 > ... < expressionn > )

```

The construct ($<\text{variable}_i>: \text{IN } <\text{set}_i>$) causes the variable to iterate over the elements of the set, which may be specified as a list, a vector, or a matrix row or column. Unless a false when-expression is present, the FOR-body is evaluated and a result is produced as governed by the FOR-keyword. Iteration then proceeds to the next set of variable values.

FOR keywords

:ALWAYS	true if all the values of body are true
:FILTER	produce a list of the non-NIL values of body
:FIRST	produce the first non-NIL value of body
:SAVE	produce a list of all values of body.

While the description of these constructs is procedural in form, the effect when programming in this notation is that of writing FORALLs and EXISTS, with the proviso that any variable values that are found to "EXIST" are collected in accordance with the FOR keyword and returned as value. The following section contains an example application of LIMAP.

DRAPHYS in LIMAP

Figure 2 summarizes the operation of the DRAPHYS diagnostic system.

```
for each C in set-of-components do
    if "C has failed" is a valid hypothesis
        then add C to set-of-valid-hypotheses;
end for;
```

component C is a valid hypothesis iff there is a POSSIBLE PROPAGATION PATH from C to every symptomatic sensor

A path is a POSSIBLE PROPAGATION PATH iff every instrumented component on the path has at least one symptomatic sensor

```
If set-of-valid-hypotheses contains
one element: done
more than one element: DRAPHYS waits for more symptoms to develop and
disambiguate the diagnosis.
```

Figure 2: Basic DRAPHYS Operation

LIMAP allows this procedure to expressed concisely. The code shown in Figure 3 creates the set of valid hypotheses:

```
(defun determine-hypotheses (components symptomatic-sensors)
; components =(def) set of all components to be
; considered as hypotheses
(for (c :in components)
    :when (is-valid-hypothesis c) :filter c))

(defun is-valid-hypothesis (c symptomatic-sensors)
(for (s :in symptomatic-sensors)
    :always (exists-bad-path c s) )

(defun exists-bad-path (c s)
(for (p :in (paths 'engine c s)) ; paths from c to s
    :first (for (component :in p) :always (not-known-ok component)) ) 

(defun not-known-ok (c)
(or (null (instrumentation c)) (symptomatic c))) ; symptomatic is a boolean vector

(defun instrumentation (c) ; returns list of sensors associated with c
(for (s :in components)
    :when (and (is-sensor s) (retrieve 'engine c s)) :save s))
```

Figure 3: DRAPHYS in LIMAP

The Size Barrier

The key statement in the Figure 3 code is the line

```
(for (p :in (paths 'engine c s)) ; paths from c to s
```

occurring in the *exists-bad-path* function. (*paths 'engine c s*) is a reference to the PSAM for relation *paths*, and exemplifies the capability to quantize over paths that is a major strength of LIMAP. It is also the source of a combinatorial explosion in terms of storage requirements when system size grows. Whereas DRAPHYS precomputes no paths, and thereby incurs unacceptable runtimes for large problems, LIMAP precomputes all paths, and encounters space limitations as the domain size grows. In particular, replacing DRAPHYS' backtracking path search with LIMAP's PSAM capability on the 23-component jet engine model originally operated on by DRAPHYS significantly improved the run time performance of the diagnostic reasoner. Increasing the size of the model to over 100 components, however, produced a PSAM matrix that, despite its sparse representation, was so large that paging overhead made its use computationally infeasible. It was evident that precomputing and storing all possible paths between nodes was unworkable in terms of space. An approach that avoided both the inordinate time requirements of DRAPHYS and the excessive space consumption of LIMAP was required.

Node Matrices

The solution that was developed was to modify the extended Warshall's Algorithm depicted in Figure 1 so that $M[i, j]$ would store not the entire set of all paths from node i to node j , but only the nodes occurring on those paths.

Let M be an $N \times N$ node matrix, i.e., a matrix each of whose entries contains a set of nodes. Initially $M[i, j] = \{i, j\}$ i.e., the set containing the two nodes occurring on the one-step path from i to j , iff there is a link from node i to node j ; if not, $M[i, j] = \text{NIL}$. Then after the loop in Figure 4 is executed, $M[i, j]$ contains the set of all nodes occurring on paths from node i to node j . More precisely: after executing the node matrix computation algorithm, $M[i, j] = \{n \mid \exists \text{path } p \text{ from node } i \text{ to node } j, \text{ and } n \text{ occurs on } p\}$

```
for k := 1 to N do ; scan array from top down
  for i := 1 to N do ; scan array from left to right
    if (i ≠ k & M[i, k] ≠ NIL) then
      for j := 1 to N do M[i, j] := M[i, j] ∪ (M[i, k] ∪ M[k, j])
```

Figure 4: Node Matrix Computation

It is important to note that if it is necessary to establish only the *existence* of a path between i and j , the node matrix M is as efficient as the adjacency matrix: a path exists iff $M[i, j]$ is not NIL. If an actual path between nodes i and j is required, the original backtracking search method is employed, but with the constraint that the search consider

only nodes in the set $M[i, j]$. If the net is sufficiently sparse (as is the case in our application), it is feasible to generate the set of all possible paths between i and j by the same approach.

Performance Improvement

The time required to compute the set of all paths between two nodes of a digraph is known to be exponential in the number of nodes. The extended Warshall's Algorithm employed by LIMAP is $\mathcal{O}(n^5)$, while the node matrix computation is $\mathcal{O}(n^4)$. The amount of time saved was found in practice to increase the size of the problems that could be attacked by nearly an order of magnitude. At least as important, however, is the fact that restricting the search to nodes in $M[i, j]$ can achieve a radical reduction in the size of the search space, depending on the degree of sparseness of M . Since a node matrix records only the set of nodes occurring on any path between i and j , rather than the paths themselves, it is to be expected that node matrices require significantly less storage than path matrices. A further source of performance improvement results from the fact that sets allow the familiar representation in terms of bit strips (bit i is on iff i is in the set) that allows set union to be implemented as bitwise OR, a highly efficient operation on essentially all machines. Each matrix entry is then a single binary number of N bits. Not only matrix storage requirements but also matrix creation time are greatly reduced compared to the PSAM, since the time-consuming link operation is no longer needed. Furthermore, if the net is sparse, then most bits of most bit strips will be 0, allowing a number of well-known compression techniques to be applied to the entries of the node matrix. The amount of storage required is $N^2 \cdot s \cdot k$ bits, where s is the expected number of distinct nodes on paths between arbitrary nodes, and k is compression overhead. In the worst case (no compression) the amount of space required is $N^2 \cdot N \cdot \log_2(N)$ bits. We thus have representational parsimony on two levels: the sparse-matrix representation facilities afforded by LIMAP, and the bit strip compression technique employed at the level of the matrix entries. Table 1 summarizes the time and space savings achieved by the node matrix technique⁴.

Model	LIMAP	Node Matrix
23-component engine:	Time: nominal Space: nominal	Time: nominal Space: nominal
166-component hydraulic system:	Time: 5 hours Space: 3M	Time: 30 minutes Space: 50K
198-component oil system:	Time: 12 hours Space: 5M	Time: 35 minutes Space: 61K
490-component fuel system:	Attempt abandoned; infeasible due to paging problems	Time: 1 hour Space: 65K

Table 1

⁴ The "nominal" entries for the 23-component engine model reflect the fact that most circumstances tend not to be noticed - or measured - until they become irksome.

It is apparent that the task addressed by DRAPHYS is a typical FED problem, and that the node matrix technique we have described is applicable to FED representations in general. LIMAP, which was developed to accommodate the requirement for quantifying over paths as well as individuals in FED representation, runs into space limitations the space limitations illustrated by Table 1 when processing larger models. If (as is frequently the case) the net is sparse, the node matrix technique allows significantly larger systems to be represented in feasible amounts of space, while retaining sufficient speed to allow quantization over paths. The example graph (tree, in this case) depicted in Figure 5 illustrates the striking reduction in search effort that the node matrix technique can achieve.

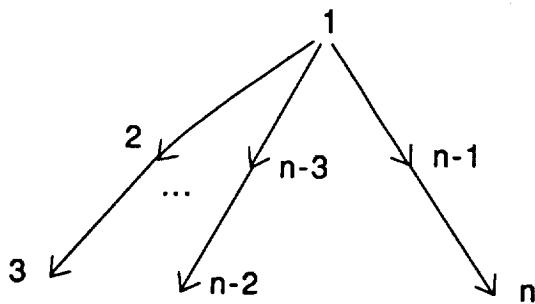


Figure 5

Suppose the task is to find a path from node 1 to node n . It is evident that depth-first backtracking search will explore all of nodes 1 to n . If a node matrix is used, then the $[1,n]$ entry will contain $(1, n-1, n)$, the set of nodes occurring on the path(s) from 1 to n . Search in this 3-node space is trivial. While this is admittedly an extreme example, we have found that the use of a node matrix increases the size of the problem that can feasibly be attempted by nearly an order of magnitude.

Conclusion

An earlier paper (Feyock and Karamouzis, 91) described the LIMAP system that we developed in response to the need for a higher-order logic capability in FED problems, particularly the need to quantize over relations and paths in diagnostic systems. Practical experience with this system showed that while it coped well with models of small to moderate size, large representations resulted in unacceptable storage requirements. In this paper we have presented technique that can lead to drastic reductions in search space size, allowing models of more than 500 nodes to be processed. The procedure described is applicable not only in the context of LIMAP, but to a number of other combinatorially explosive search space problems found in AI as well.

References

Abbott, K. H. *Robust Fault Diagnosis of Physical Systems in Operation*, NASA Technical Memorandum 102767, NASA Langley Research Center, Hampton, Virginia, 1991.

Charniak, E., et al. *Artificial Intelligence Programming* (2nd Ed.). Lawrence Earlbaum Associates, 1987.

Feyock, S., and S. Karamouzis, "Design of an Intelligent Information System for In-Flight Emergency Assistance," in *Proceedings of the 1991 Goddard Conference on Space Applications of Artificial Intelligence*, Greenbelt, MD, May 1991.

Feyock, S., and S. Karamouzis, "A Path-Oriented Matrix-based Knowledge Representation System," in *Proceedings of the Fourth International Conference on Tools with Artificial Intelligence*, Arlington, VA, November 1992.

Horowitz, E., and Sahni, S. *Fundamentals of data structures*. Computer Science Press, 1976.

1995120975

N95- 27396

Real-time Value-driven Diagnosis

Bruce D'Ambrosio
Department of Computer Science
Oregon State University
dambrosi@research.cs.orst.edu

521-63

47271

P-16

Abstract

Diagnosis is often thought of as an isolated task in theoretical reasoning (reasoning with the goal of updating our beliefs about the world). We present a decision-theoretic interpretation of diagnosis as a task in practical reasoning (reasoning with the goal of acting in the world), and sketch components of our approach to this task. These components include an abstract problem description, a decision-theoretic model of the basic task, a set of inference methods suitable for evaluating the decision representation in real-time, and a control architecture to provide the needed continuing coordination between the agent and its environment. A principal contribution of this work is the representation and inference methods we have developed, which extend previously available probabilistic inference methods and narrow, somewhat, the gap between probabilistic and logical models of diagnosis.

1 Introduction

In this paper we will present a status report on research into diagnosis as an embedded value-driven activity¹. Several characteristics become apparent from this perspective that are not normally emphasized in work on diagnosis. First, we view the task as on-going rather than one-shot. Second, we recognize that diagnosis must often be performed under time limitations. Finally, we consider a decision-theoretic model of diagnostic activity in which diagnostic reasoning is not a theoretical activity but a practical one. That is, its goal is not to increase our knowledge, but rather to choose an action which maximizes expected utility.

Four key elements characterize our current approach to these problems. First, we have developed a simple abstract domain which captures these essential elements, which we title the On-Line Maintenance Agent (OLMA) task domain. Second, we have developed a decision-theoretic model of the basic decision task facing an agent in this domain. Third, we have developed probabilistic inference methods suited for solving the resulting decision problem. Fourth, we have developed a problem-solving architecture intended to provide the reactivity needed in embedded, ongoing tasks. We view the third contribution, the development of suitable probabilistic inference mechanisms, as our most significant contribution to date. These methods have been described elsewhere. The goal of this paper is to place them in the context of the larger task of real-time value-driven diagnosis. We begin with a review of the basic task domain, then discuss each of the four elements of our approach. We close with a discussion of related work, limitations, and future directions.

¹Acknowledgment: This work supported by NSF 91-00530 and the Oregon Advanced Computing Institute.

2 On-Line Equipment Maintenance

Early writings by at least some in AI (e.g., [26]) stress the necessity of recognizing both the limitations and embedded nature of realizable agents. Despite that, much work in AI has proceeded as if the demands placed by the environment are static. In a diagnostic task, for example, the system being diagnosed is often presumed to have broken before diagnosis starts, and to not undergo any further changes (other than those initiated by the diagnostic agent) while diagnosis takes place (an exception is some medical monitoring and diagnosis systems). Further, the environment is presumed to be static enough to allow all solution-quality/time-to-solution tradeoffs to be made at design time (the standard "20 minute solution time" criterion for judging potential expert system applications is a prime example of this). These assumptions, together with assumptions that problem solving proceeds by inference with explicit, categorical representations, characterize much work in AI, whether it is on planning, diagnosis, natural language understanding, or image understanding (although less so in these latter areas).

Diagnosis, in accord with the above, is often formulated as a static, detached process, the goal of which is the assessment of the exact (or most probable) state of some external system. In contrast, we view diagnosis as a dynamic, practical activity by an agent engaged with a changing and uncertain world. We have focused our initial investigations of diagnosis on the task of diagnosing a simple digital system *in situ*. Our formulation of embedded diagnosis has the following characteristics:

- The equipment ² under diagnosis continues to operate while being diagnosed.
- Multiple faults can occur (and can continue to occur after an initial fault is detected).
- Faults can be intermittent.
- There is a known fixed cost per unit time the system is malfunctioning.
- The agent senses equipment operation through a set of fixed sensors and one or more movable probes.
- Action alternatives include probing test points and replacing individual components. Each action has a corresponding cost.
- The agent can only perform one action at a time.
- The overall task is to minimize total cost over some extended time period during which several failures can be expected to occur.

We term this task the *On-Line Maintenance* task, and an agent intended for performing such a task an *On-Line Maintenance Agent*. An interesting aspect of this reformulation of the problem is that diagnosis is not a direct goal. A precise diagnosis is neither always obtainable nor necessary. Indeed, it is not even obvious *a priori* what elements of a diagnosis are even relevant to the decision at hand.

²We will use "system" to refer to our diagnostic system, and "equipment" to refer to the target physical system.

3 A decision-theoretic formulation of an OLMA

Our first commitment is that the task is essentially a decision-theoretic one. That is, the essential task of the agent is to *act* in the face of limited information. In order to formulate this problem decision-theoretically, the agent must have knowledge of several parameters of the situation: It must know the cost of each type of replacement or probe act, the cost of system outage, and expected probabilities of component failures over the next decision cycle³. A naive attempt to formulate this task decision-theoretically, however, encounters three problems. First, a proper decision-theoretic consideration of this task would require looking ahead over all decisions over the entire operational life of the equipment in order to optimize the first decision. This is clearly computationally intractable. Second, even if the first problem can be solved, time is passing while the agent is computing the first action, and it is not clear how the agent should trade quality of a decision for timeliness of the solution in choosing actions. Finally, the agent must act repeatedly, yet each action is in a new context: not only must a new set of input data be considered, but also a new set of beliefs about system state, based on prior information and computation.

The infinite lookahead, or "small worlds" [23] problem has two subproblems, one for replacement actions and another for probe actions. We circumvent the first subproblem, that of infinite lookahead for replacement actions, as follows. For replacement actions it is possible, under assumptions about stability of the fault transition probabilities and fault costs, to derive off-line an optimal replacement policy conditioned on current beliefs about component states (this would take the form: "replace component x whenever its posterior probability of being faulted is greater than p_x ," where p_x depends on parameters of the target system). We use an alternate, equivalent form of this result that requires less a-priori analysis, an assumption of policy stability. This assumption is roughly as follows: If I choose not to replace a component now, then, all other things being equal (ie, no new unexpected sense data), I will make the same choice next time. Under this assumption, the temporal consequences of a decision extend, not for a single sense/act cycle, but several decision cycles into the future. We model the temporal extent of the outcome state for a decision as fixed time-period chosen to satisfy the following constraints:

$$t >> r/f$$

$$t << r/pf$$

where:

t is the outcome state duration (effectively, the multiplier for failure costs),

r is the cost of component replacement,

f is the cost of component failure for a unit clock time, and

p is the probability of component failure during a unit time interval.

The first constraint arises from the observation that, if the cost of replacement is greater than the cost of failure over the outcome period, then our limited look-ahead agent will never

³These latter two costs will vary with agent processing capacity, since a slower agent will take longer to make a decision. This will increase the chance of a component failing during a single decision cycle, and increase the cost of a system outage over a decision cycle.

replace a component. The second constraint arises from a more subtle problem: even if we are certain the equipment is functioning correctly now, there is a non-zero probability it will be in failure mode in the outcome state. If the expected cost of that failure ($p * f * t$) is greater than component replacement cost (r), then the agent will always choose to replace, even though it believes the equipment is functioning perfectly. Note that the requirement that both constraints be satisfied limits our approach to equipment with relatively low failure rates.

We resolve the second subproblem, that of determining the expected value of probe actions, by using the standard decision-theoretic heuristic of one-step look-ahead. The result of these two techniques gives us an abstract decision-basis for the first decision as shown in figure 1, where the link from the state at time zero to the value node reflects the costs of operating in that state for one decision time, and the link from the state at time one to the value node reflect the cost of operating in that state for t time units.

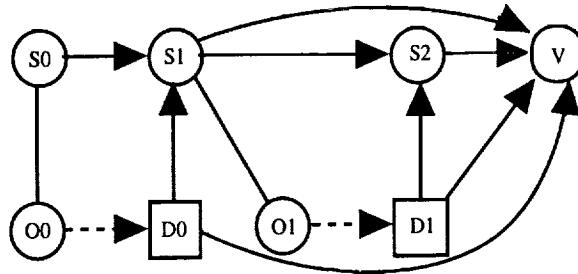


Figure 1: Abstract Decision Basis for first decision

A few notes about this representation are in order. S_0 , S_1 , and S_2 represent the state of the equipment at times 0, 1, and 2 respectively. The actual state values are unknown to the agent. O_0 and O_1 represent the observational data available to the agent at the time decisions D_0 and D_1 are to be made. The solid arcs from S_0 and S_1 to O_0 and O_1 respectively reflect the fact that O_0 and O_1 are directly influenced by the (unobservable) system state. The dashed arcs from O_0 to D_0 and O_1 to D_1 represent the fact that the actual values of these variables will be available at the time the respective decisions will be made. Finally, V represents the value of possible outcomes, and the arcs to it reflect that the value is a function of the actual state of the equipment at times 0 and 1 as well as the direct costs of the actions the agent chooses.

We refer to this as an “abstract” decision basis because we typically will not represent the state of the system or the observational data as single state variables. Rather, we will typically have a structured representation, in the form of a belief net. S_0 , for example, represents the set of unobservable system state parameters and the relationships among them, O_0 represents the set of observed parameters, and the arc between S_0 and O_0 represents the relationship between the unobservable and observable system parameters. This general model can be instantiated with causal or evidential relationships (or both) between unobservable and observable parameters. Consider, for example, a simple sequential two inverter circuit. For that case, we might use causal knowledge to build a specific instantiation of our abstract S_0 and O_0 as shown in figure 2. In this figure, “Inv1, P, and Inv2” might all be components of “ S_0 ”, and “I and O” might be directly observable.

Our second problem was that of trading off quality of solution with time to solution. There are two issues here. First, if the equipment is faulted, the longer we delay taking a repair action, the higher the cost incurred. Second, since equipment operation is in parallel with

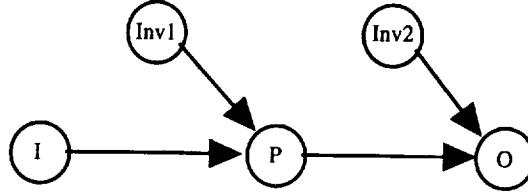


Figure 2: Decision Basis fragment for two inverter system

agent operation, a fault may occur *while* the agent "wastes" time reasoning about a prior, correct set of sense data. We provide no solution to either of these problems here. However, we do present later an incremental (anytime) decision procedure, and sketch experiments we are performing to statistically infer effective reasoning policies.

We resolve our final problem, that of making subsequent decisions, by simply extending the above decision basis forward in time by one decision each cycle. A sample decision basis for the second decision made by the maintenance agent is shown in figure 3. This method would seem to have a problem: one would expect that decision time (and space) would increase at least linearly with time. In fact, both time and space requirements are constant. We defer the discussion of how we accomplish this to the next section, on incremental probabilistic inference.

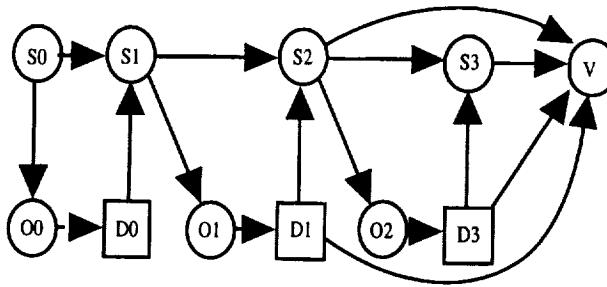


Figure 3: Abstract Decision Basis for second decision

In summary, then, our base-level agent executes the following cycle each time it is called upon to choose an action:

1. Extend the decision basis forward in time by one decision cycle.
2. Acquire current sense data (including probe values).
3. Find the action with minimum expected cost.
4. Post the selected act as evidence in the belief net, prune unneeded old cycles from the net, and return selected action.

4 Incremental Probabilistic Inference

The problem of computing the expected utility for action alternatives can be cast as a belief-net inference problem, as shown by Cooper [3]. However, most current belief net inference facilities provide poor support for the cycle described above. The limitations in existing algorithms are

both representational and inferential. The key representational limitation is an inability to represent structure within a single conditional probability distribution. The key inferential problems can be described as lack of *incrementality* with respect to various task requirements.

4.1 Representation

A belief net [21] is a compact, localized representation of a probabilistic model. The key to its locality is that, given a graphical structure representing the dependencies (and, implicitly, conditional independencies) among a set of variables, the joint probability distribution over that set can be completely described by specifying the appropriate set of marginal and conditional distributions over the variables involved. When the graph is sparse, this will involve a much smaller set of numbers than the full joint. Equally important, the graphical structure can be used to guide processing to find efficient ways to evaluate queries against the model. For more details, see [21], [5], [24], [19]. All is not as rosy at it might seem, though. The graphical level is not capable of representing all interesting structural information which might simplify representation or inference. The only mechanism available for describing antecedent interactions in typical general purpose belief net inference algorithms is the full conditional distribution across all antecedents. However, a number of restricted interaction models have been identified which have lower space and time complexity than the full conditional. The noisy-or [21], [22], [13], [14], [1] for example, can be used to model independent causes of an event, and inference complexity is linear in both space and time in the number of antecedents for many inferences. Similarly, various asymmetric [25], [12] and logical relationships are inefficiently represented using a full conditional. Finally, value models used in utility modeling are often factored, for example they may be additive. We have developed an algebraic extension to belief nets which permits conditional distributions to be defined as algebraic compositions of smaller distributions. We have shown that this representation is capable of capturing all known intra-distribution structures, and that simple inference algorithms can make use of this structure to perform inference effectively. Consider, for example, the problem of modelling the conditional distribution of the output of our two inverter circuit given the state of the second inverter and the output of the first inverter. This would normally be modeled as a single monolithic conditional distribution containing (if the inverter has four states: Ok, Stuck-at 0, Stuck-at 1, and Unknown) 16 numbers. The fact that three of the four inverter states are deterministic is lost. However, we model this as follows using our local expression language:

$$\begin{aligned} \exp(O) = & P(O_1|Inv2 = Ok, I1out_0) \\ & + P(O_0|Inv2 = Ok, I1out_1) \\ & + P(O_0|Inv2 = S0) + P(O_1|Inv2 = S1) \\ & + P(O|Inv2 = Uk) \end{aligned}$$

A total of only six numbers are needed for the above (4 of them are 1.0). But more significantly, we have captured explicitly important structural information such as the fact that the output is independent of the input when the inverter is Stuck-at 1, Stuck-at 0, or Unknown, and that inverter behavior is deterministic in the Ok, Stuck-at 0, and Stuck-at 1 states.

We believe this representation goes a long way towards closing the expressiveness gap between probabilistic and propositional representations for device models.

4.2 Inference

Now that we have an adequate abstract model for the basic on-line maintenance decision and a representation adequate for expressing device and utility models, we need inference methods suited for the task at hand. Despite the fact that current state-of-the-art algorithms exploit the independence information in a belief net to construct efficient computations for probabilistic inference [21], [17], [24], in practice computational cost still grows rapidly [18], limiting application of these techniques to belief nets with a few hundred variables. Also, the services offered by current probabilistic reasoning systems are not well matched to the needs of higher-level problem solvers. As we discussed in [4] and [6], problem solvers typically interleave model construction, revision, and evaluation. As a specific example of this, consider the requirements posed by our formulation of the OLMA task:

- We must extend a model forward in time without discarding all previous inference and starting over. Current methods lack the incrementality with respect to model reformulation operations needed to support this.
- Our decision evaluation methods require that we query arbitrary subsets of the network variables. Current methods lack the incrementality with respect to queries needed to support this.
- The embedded nature of the task requires that we have flexible methods for performing partial inference. Current methods lack the incrementality with respect to completeness needed to support this.

In summary, no existing general-purpose low-level (propositional) probabilistic representation service provides incrementality with respect to model revision and resource usage in a theoretically sound manner. We, of course, have an answer. In this subsection we begin by reviewing the inference problem in a more general setting, and offering a redefinition of the basic inference task. We sketch how an inference engine which performs this task can serve as the core of an incremental probabilistic representation service, and report on the status of our current implementation.

4.2.1 Term Computation

Probabilistic inference in belief nets, as currently defined, is generally taken to be the computation of the prior or posterior marginal, conjunctive, or conditional probability distribution over some subset of variables in a fixed network. This is often an unnecessarily restrictive formulation of the problem. The actual computation of any prior or posterior probability can in general be viewed as a sum over a number of terms (in the extreme case, this occurs as marginalization of the full joint). While the number of terms to be computed is exponential in the size of the network, the time complexity of computation of a single term is linear. Consider the network shown in figure 4.

In this network:

$$\begin{aligned}
 P(C_t) &= \sum_{A,B} P(C_t|B, A) * P(B) * P(A) \\
 &= .8 * .8 * .8 + .8 * .2 * .2 + .2 * .2 * .8 \\
 &\quad + .2 * .2 * .2 \\
 &= .512 + .032 + .032 + .008
 \end{aligned}$$

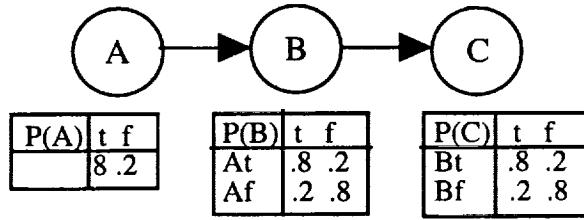


Figure 4: Simple Belief Net

We take the computation of a single term as an appropriate primitive task for probabilistic inference, and next show how an inference system with the needed incrementality properties can be built around it.

4.2.2 Desiderata for a TCS

A term computation approach will be interesting only if we can get a significant amount of information through the computation of a small number of terms. While there are many ways this might arise⁴, we motivate the approach through the introduction of a critical assumption: we assume that most distributions in a belief net are “asymmetric.”

Definition 1 *A marginal probability distribution is Asymmetric if one mass element is larger than the remaining element(s). A conditional distribution is Asymmetric if each row satisfies this constraint. In this case it need not be the same element in each row.*

If all the distributions in a belief net are asymmetric, then most of the probability mass for many queries is contained in the first few terms⁵:

Theorem 1 *Given a Belief-net over n two-valued variables such that all distributions are asymmetric with a larger mass at least $(n - 1)/n$, then the $n + 1$ largest terms in the joint distribution across the variables contain a total mass of at least $2/e$.*

Note that this result is not based on any assumptions about the structure of the network. The degree of asymmetry assumed in the above theorem may seem extreme. However, it is quite natural in many applications, such as failure modeling of complex systems. Thus, our answer to the question of which terms to compute will be to compute the largest terms first.

One could construct a term computation system which merely enumerated elements of the full joint distribution across all variables in a network, as in our example. Indeed, existing proposals for anytime probabilistic inference essentially do this [15], [16]. However, such an approach can be grossly inefficient. There are several sources for this inefficiency: First, there would be a time inefficiency due to unnecessary repetition of sub-computations (eg, the computation of $P(B_t|A_t) * P(A_t)$ in our example). Second, there would be space inefficiency resulting from keeping each term separate. Finally, it is not obvious how such simple methods can be made incremental with respect to newly arriving evidence, queries, or belief net extensions.

Developments in exploiting the probabilistic independence relations expressed in belief nets provide the necessary basis for designing computations which address these problems. In general, the sparser a belief net, the more finely any computation can be partitioned into

⁴For example, through domain dependent knowledge of paradigmatic “cases”.

⁵Proof in extended report.

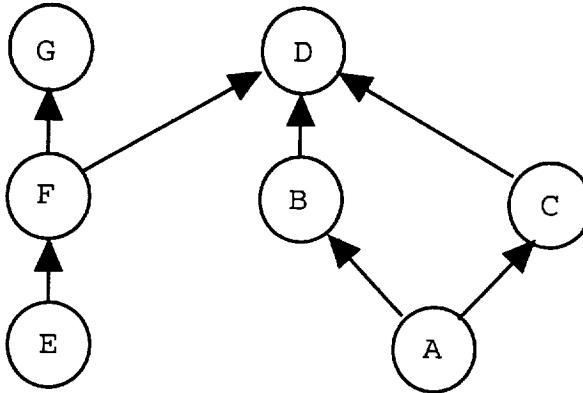


Figure 5: Paradigmatic Belief Net

independent sub-computations which share only a small number of variables. For example, given the net in figure 5, a query for $P(D)$ can be computed by first computing the full joint probability distribution, then marginalizing over all variables except D :

$$P(D) = \sum_{A,B,C,E,F} P(D|B,C,F) * P(F|E) * P(E) \\ * P(B|A) * P(C|A) * P(A)$$

However, a much more efficient form of the computation is:

$$P(D) = \sum_F (\sum_E P(F|E) * P(E)) * \sum_{B,C} P(D|B,C) \\ * \sum_A P(B|A) * P(C|A) * P(A)$$

Having done this, we can eliminate redundant computation simply by caching intermediate results. Similarly, we can reduce the space requirement by combining terms when their bindings differ only on variables not needed in the remainder of the computation. In the extreme, each of these can reduce the corresponding complexity (time and space) for computing each term beyond the first from n to $\log(n)$, where n is the number of variables relevant to a query.

In the following section we first develop the basics of term computation (which is inherently incremental with respect to resource consumption) for a static network, set of evidence, and set of queries. We then sketch how the fundamental computation can be made incremental with respect to queries, evidence, and net extension.

4.2.3 Basics of term computation

The elementary primitive out of which we build a term computation system is the construction of a stream of terms for some node in the evaluation poly-tree for a set of queries. This stream will be constructed, recursively, by combining streams of terms from child nodes in the poly-tree. We first describe this evaluation poly-tree and its construction, then explain the term computation process.

Evaluation poly-tree construction Consider the net in figure 5, and assume our queries are for G and D . The expression for $P(D)$ has been given earlier. The expression for $P(G)$ is:

$$P(G) = \sum_F P(G|F) * (\sum_E P(F|E) * P(E))$$

It should be obvious that we can efficiently combine these two expressions into a single evaluation poly-tree:

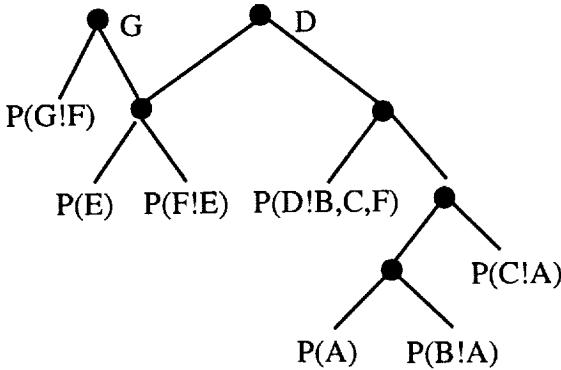


Figure 6: Evaluation Poly-tree for sample query set

Construction of an optimal evaluation poly-tree for an arbitrary query set is a hard problem [19]. However, simple, polynomial-time heuristics perform quite well, and are described in [9], [19]. This previous work was performed in the context of exact query evaluation (that is, computation of all terms), but the theory remains applicable, and so will not be repeated in detail here. The process used in [19] is the basis for our current implementation. That method is essentially a greedy search through the space of partial evaluation trees⁶.

Term computation Given an evaluation poly-tree for a query set, the primitive operation at each node in the tree is generation of a term. Term generation is simple: each term is generated by forming the product of a term from the left child and a term from the right child. There are, however, several issues to consider: (1) Control: computations can be performed in either a data-driven or goal-driven fashion; (2) Term selection: the decision of which term to compute next; (3) Term combination: the detection of terms which can be combined. We discuss each of these issues in turn, then demonstrate computation of a term using our example poly-tree.

Control Most probabilistic inference systems are data driven. SPI, the system from which the TCS is derived, however, is query driven. We retain this query driven computational style in our TCS, although it could undoubtably be adapted for data-driven computation. Computation therefore starts at one of the roots of the poly-tree when the problem solver asks for refinement of the distribution for that node, and the root queries its immediate children for terms as needed.

⁶In fact, the situation is slightly more complex when the use of local expressions is considered, since then the set of variables needed below a node in the poly-tree varies depending on the values of the variables instantiated so far. For this reason we actually interleave poly-tree construction and search.

Term selection We earlier stated that we would attempt to minimize the number of terms computed by computing largest terms first. We use AO* to search for the largest term. AO* requires two measures, a measure of “distance so far” and a heuristic estimate of remaining distance. We use the mass computed so far as the inverted “distance traveled so far,” and the partial value returned by a partial subterm as our heuristic estimate. This is an admissible heuristic, and so guarantees that the largest term will be in front of the agenda upon termination⁷. Problem solver guidance can be provided in the form of a “scaling function” which has access to term bindings and can scale the probability masses before they are used to order the search agenda.

Term combination (marginalization) The number of terms computed in response to any query is exponential in the number of relevant variables. However, the major advance offered by recent developments in probabilistic inference is reduction of the exponent for computation of complete distributions from number of relevant variables to number of relevant variables in a single factor or cluster, as we discussed earlier. We should not have to pay a higher price simply to achieve incrementality. We can achieve this efficiency by merging completed terms which are distinguished only by bindings on variables not needed at higher levels of the evaluation poly-tree. This creates two problems. First, a term which has already been incorporated into streams at higher levels in the evaluation poly-tree can suddenly have its value change (positively). Simple dependency tracking mechanisms suffice to record the information needed to update these higher terms. Second, exactly what does the AO* guarantee now mean? In poly-tree nodes where marginalization takes place, a partial term can be extended in two ways: by multiplying its value by terms from remaining distributions, or by adding additional ground terms⁸. While we use a heuristic which is admissible in its estimate of the effect of the former, our heuristic is inadmissible with regard to the latter (because it ignores marginalization). This means we can only make a relatively weak statement about terms in streams generated from poly-trees containing marginalization: that the first term returned will be that term whose lower bound is highest after considering all complete ground subterms computed so far. Note that the term need not be “complete” in the sense that further ground terms may be added into it during later computation. It is, however, complete in the sense that it is a sum of a set of complete ground terms.

Complexity The key assumptions we make are that: (1) the probability distributions are sufficiently asymmetric and; (2) the graphical structure of the belief net is sufficiently sparse. Under these assumptions, the evaluation poly-tree will be such that the total number of terms computed in all streams, in the course of computing the first n term requests for each query in the query set, will be n times the number of nodes in the poly-tree. Since the poly-tree is a binary tree, this in turn is $2n$ in the number of variables relevant to the query set. All the operations we have described are either constant time, linear, or at worst $n\log(n)$ (reordering the agendas) in the number of terms in an agenda. Therefore, the total complexity, in the admittedly most optimistic case, is $2n^2\log(n)$ where n is the number of variables relevant to a query set and the number of terms requested. Our experience in actually applying this procedure to three tasks, computation of marginal probabilities, most likely composite hypotheses,

⁷This selection criterion is similar to the techniques used by deKleer [11] and Henrion [15]. Both use search on restricted classes of networks for the diagnostic task of finding most likely composite hypotheses, with good results. One contribution of our work is to show how this technique can be used in a more general setting.

⁸A “ground” term is one with a unique binding for each variable in the subtree rooted by the poly-tree node under consideration.

and complete decision analysis, confirms that this estimate is in fact realistic for a typical class of belief nets describing decision models for diagnosis and control of simple digital circuits.

4.2.4 Making Term Computation Incremental

The basic process sketched above is incremental with respect to computation of additional terms for a static query set. We have made this process incremental with respect to queries, evidence, and limited model reformulations (namely the extension and pruning operations needed to support the OLMA task). We omit discussion of these issues here due to lack of space. For a more complete discussion see[7].

4.2.5 Discussion

We have sketched a process which is basically little more than heuristic search for the set of bindings across a set of variables that maximizes the posterior probability across those variables. In another context, deKleer has referred to this as the “Most Likely Composite Hypothesis” problem [10], Henrion has described an algorithm for diagnosis in very large knowledge bases [15], and Pearl has discussed the problem of “Distributed Revision of Composite Beliefs” [20]. From another perspective, Horvitz *et al* have been developing bounded conditioning as an approach to anytime probabilistic inference [16], and Boddy has proposed an anytime approach to dynamic programming [2]. We believe the contributions of our work are several: (1) We have shown how, with caching and marginalization, an incremental probabilistic inference system based on computation of individual terms can be made as efficient at computing all terms (within a factor of $\log n$) as the best algorithms for exact inference; (2) We have demonstrated that this process can be made incremental with respect to queries, evidence, and model revisions; (3) We have argued that such a system can serve as the basis for a tractable general-purpose low-level representation service. In particular, this general formulation can be used for evaluation of decision bases, as will be discussed in the next section.

4.3 Value-Driven Diagnosis

There is still a missing link in our base-level OLMA story. What does term computation have to do with evaluating a decision basis, and what does that have to do with diagnosis? Our hypothesis is that the theorem we presented for general *asymmetric* belief nets holds for typical decision bases. Further, we make a stronger hypothesis: that the computation of the few largest terms in the expected utility will be sufficient to distinguish the highest expected utility action. An expected utility is the sum of a number of terms, each of which is a probability of the system resting in a certain outcome state (given the local expression language, perhaps a partial state) multiplied by the (partial)⁹ utility of that (partial) state. Our hypothesis will be valid, then, when most terms will be for either low probability outcomes or low utility aspects of an outcome. In our system, then, diagnostic reasoning is driven by the search for high payoff (either positive or negative) utility terms, which drives a search for high probability outcomes, which in turn drives a search for high probability current state information (note that, due to the factoring which takes place in the evaluation poly-tree construction and the structure made explicit in the local expression language, it is not obvious to state a priori what current state information will be queried during expected utility term computation.).

⁹As a result of the additive value model.

Does all this work? We have only executed the system on small digital circuits to date, due to high overhead in our prototype implementation. However, we have observed linear term computation times over a range of 1 to 16 gates. By contrast, exact evaluation of the decision basis exhausts available space (90mb swap on a Sparcstation 1+) at 8 gates. Figure 7 is a typical performance graph. The task is monitoring a simple 4 gate "half-adder." The vertical axis is cost-per-failure (total scenario cost divided by number of failures), and the horizontal axis is the number of cpu seconds corresponding to one "tick" of the simulation clock (thus, higher numbers simulate a faster cpu for the agent). The solid dots record the cost of running an exhaustive decision algorithm, the crosses record the cost using the TCS. These preliminary results indicate the TCS approach is considerably more robust with respect to task timeliness requirements than is exact computation.

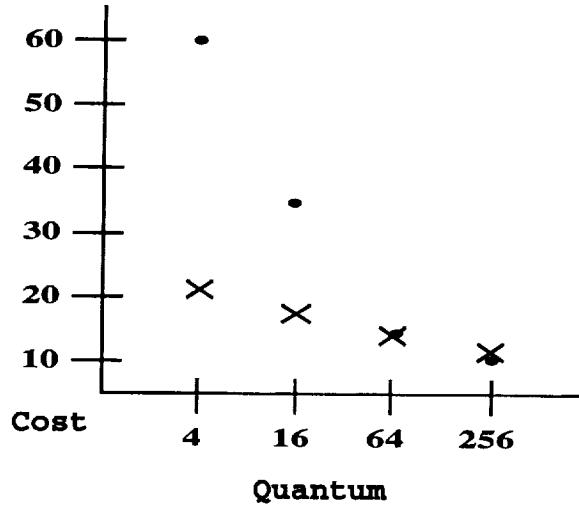


Figure 7: Performance of TCS vs Complete eval

5 Reaction and deliberation in real-time diagnosis

Each TCS point in the above graph is obtained by first performing a series of runs to determine the optimal number of terms to compute for a given cpu speed, then measuring the cost using that number of terms. Thus, the number of terms computed is higher for quantum 256 (1024 terms) than for quantum 1 (1 term). Table 8 shows the optimal number of terms for each quantum.

This presumes that the number of terms computed on each decision cycle is constant and independent of both internal and external state. It further presumes the agent will ignore any sense data that arrives while a decision computation is in progress. This is a very naive approach to real-time problem solving. More generally, we can imagine a meta-decision process which is invoked whenever either: (1) new sense data arrives; or (2) the state of decision evaluation changes (eg, completion of the computation for an additional term). Whenever a state change occurs in either, this meta-decision process has the following options: take an action directly, initiate, terminate, or abort a decision evaluation, or push a new decision evaluation on the stack of current evaluations. Information available for this decision includes the current sense data and the current state of the computation (number of terms computed

Quantum	Optimal Number of Terms	Avg Cost/Failure
1	1	32.5
2	32	28.0
4	32	23.0
8	32	21.5
16	64	20.0
32	256	19.5
64	256	19.3
128	512	16.0
256	1024	14.5

Figure 8: Performance of term computation as Quantum varies

so far, most likely composite hypothesis, etc). For further details of this architecture, see[8]. We are currently applying statistical techniques identify optimal single-level control policies for the OLMA domain.

6 Discussion

It is interesting that the search technique we present should work at all in a decision-making context. There are good theoretical reasons for expecting that search should work well for diagnosis, that is, for identifying the most likely composite hypothesis across component states. The technique proposed here essentially first explores single fault hypotheses, then double fault, and so on. The extension to decision models might seem much less clear. However, recent theoretical results by Druzzel [?] show that almost all Bayesian networks, even those not possessing the local skewness properties described earlier, will contain a few dominant terms. Our experimental results seem to confirm this: The decision networks we use contain many variables whose distributions do not meet the requirements of skewness. Especially noteworthy are the decision and value nodes. Decision nodes, for example, contain no distribution at all, and so provide no search guidance. Nonetheless, our methods provide robust decision-making across a range of quality/time tradeoffs. It may be that this is in part due to the forgiving nature of the problem: there are no catastrophically bad decisions, the worst the agent can do is to merely delay in making the correct decision, and accrue a small failure penalty. It is important to note, however, that even in the worst case (only one term computed) the agent repaired every fault within a few cycles.

We are aware of several limitations in our current approach. Our model of time is quite naive and limited. Our current inability to handle continuous variables is a serious restriction. But perhaps the most severe limitation is the assumption of a complete base level decision model applicable to all situations. We are currently beginning work on both of these latter problems. Specifically, we are exploring methods by which situated decision models can be dynamically constructed from background domain theories, using a meta-level version of the same base level architecture sketched here, and using both bottom up and top-down construction methods.

7 Summary

Diagnosis is often thought of as an isolated task in theoretical reasoning. We have presented a decision-theoretic interpretation of diagnosis as a task in practical reasoning, and sketched

components of our approach to this task. The approach makes significant strides to integrating commonly held logical and probabilistic models of diagnosis, as well as incorporating a real-time element. Preliminary results indicate that the approach is significantly more robust in the face of cpu speed variations than traditional approaches.

8 Bibliography

References

- [1] M. Agosta. Conditional inter-causally interdependent node distributions ... In *Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 9–16. Morgan Kaufmann, Publishers, July 1991.
- [2] M. Boddy. Anytime problem solving using dynamic programming. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 738–743. AAAI, July 1991.
- [3] G. Cooper. A method for using belief networks as influence diagrams. In *Proceedings of the 1988 Workshop on Uncertainty in AI*, pages 55–63. Association for Uncertainty in AI, August 1988.
- [4] B. D'Ambrosio. Process, structure, and modularity in reasoning under uncertainty. In *Proceedings of the 1988 Workshop on Uncertainty in AI*, pages 64–72. AAAI, August 1988.
- [5] B. D'Ambrosio. Symbolic probabilistic inference. Technical report, CS Dept., Oregon State University, 1989.
- [6] B. D'Ambrosio. Incremental evaluation and construction of defeasible probabilistic models. *International Journal of Approximate Reasoning*, July 1990.
- [7] B. D'Ambrosio. Term computation systems. Technical report, CS dept., Oregon State University, 1991.
- [8] B. D'Ambrosio and T. Fountain. Reaction in real-time decision making. Technical report, CS Dept, Oregon State University, 1991.
- [9] B. D'Ambrosio and R. Shachter. Symbolic probabilistic inference. Technical report, Oregon State Univ. CS dept., 1992.
- [10] J. de Kleer. Focusing on probable diagnoses. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 842–848. AAAI, July 1991.
- [11] J. de Kleer and B. Williams. Diagnosis with behavioral modes. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1324–1330. IJCAI, August 1984.
- [12] D. Geiger and D. Heckerman. Advances in probabilistic reasoning. In *Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 118–126. Morgan Kaufmann, Publishers, July 1991.
- [13] D. Heckerman, J. Breese, and E. Horvitz. The compilation of decision models. In *Proceedings of the Fifth Conference on Uncertainty in AI*, pages 162–173, August 1989.

- [14] M. Henrion. Towards efficient probabilistic diagnosis with a very large knowledge-base. In *AAAI Workshop on the Principles of Diagnosis*, 1990.
- [15] M. Henrion. Search-based methods to bound diagnostic probabilities in very large belief nets. In *Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 142–150. Morgan Kaufmann, Publishers, July 1991.
- [16] E. Horvitz, H. J. Suermondt, and G. Cooper. Bounded conditioning: Flexible inference for decisions under scarce resources. In *Proceedings of the Fifth Conference on Uncertainty in AI*, August 1989.
- [17] S. Lauritzen and D. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, B* 50, 1988.
- [18] Z. Li. Experimental characterization of several algorithms for inference in belief nets. Technical report, Master's thesis, CS Dept., Oregon State University, 1990.
- [19] Z. Li and B. D'Ambrosio. An efficient approach to probabilistic inference in belief nets. In *Proceedings of the Annual Canadian Artificial Intelligence Conference*. Canadian Association for Artificial Intelligence, May 1992.
- [20] J. Pearl. Distributed revision of composite beliefs. *Artificial Intelligence*, 33(2):173–216, 1987.
- [21] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, Palo Alto, 1988.
- [22] Y. Peng and J. Reggia. A probabilistic causal model for diagnostic problem solving - part 1: Integrating symbolic causal inference with numeric probabilistic inference. *IEEE Trans. on Systems, Man, and Cybernetics: special issue on diagnosis*, SMC-17(2):146–162, 1987.
- [23] L. Savage. *The Foundations of Statistics*. Dover Publications, 1972.
- [24] R. Shachter, B. D'Ambrosio, and B. DelFavero. Symbolic probabilistic inference in belief networks. In *Proceedings Eighth National Conference on AI*, pages 126–131. AAAI, August 1990.
- [25] R. Shachter and R. Fung. Contingent influence diagrams. Tech report, Dept. of Engineering Economic Systems, Stanford University, September 1990. In preparation.
- [26] H. Simon. *Sciences of the Artificial*. MIT Press, 1974.

1995120976

47272

P. 17

A Linguistic Geometry for 3D Strategic Planning

Boris Stilman

Department of Computer Science & Engineering, University of Colorado at Denver
Campus Box 109, Denver, CO 80217-3364, USA.
Email: bstilman@cse.cudenver.edu

Abstract. This paper is a new step in the development and application of the Linguistic Geometry. This formal theory is intended to discover the inner properties of human expert heuristics, which have been successful in a certain class of complex control systems, and apply them to different systems. In this paper we investigate heuristics extracted in the form of hierarchical networks of planning paths of autonomous agents. Employing Linguistic Geometry tools the dynamic hierarchy of networks is represented as a hierarchy of formal attribute languages. The main ideas of this methodology are shown in this paper on the new pilot example of the solution of the extremely complex 3D optimization problem of strategic planning for the space combat of autonomous vehicles. This example demonstrates deep and highly selective search in comparison with conventional search algorithms.

1. INTRODUCTION

Aerospace problems such as long and short-range mission planning, especially for autonomous navigation, scheduling, aerospace robot control, long-range satellite service, aerospace combat operations control, etc. can be formally represented as reasoning about complex large-scale control systems. The field of efficient aerospace control systems needs new technology from the science of artificial intelligence (Rodin, 1988; Lirov, Rodin et al., 1988).

The classic approach based on the theory of Differential Games alone is insufficient, especially in case of dynamic, 3D models (Garcia-Ortiz et al., 1993). Following (Rodin, 1988; Shinar, 1990) discrete-event modeling of complex control systems can be implemented as a purely interrogative simulation. These techniques can be based on generating geometrically meaningful states rather than time increments with due respect to the timeliness of actions. By discretizing time, a finite game tree can be obtained. The nodes of the tree represent the states of the game, where the players can select their controls for a given period of time. It is also possible that players do not make their decisions simultaneously and in this case, the respective moves of the two sides can be easily distinguished. Thus, the branches of the tree are the moves in the game space. The pruning of such tree is the basic task of heuristic search techniques.

Interrogative approach to control problems offers much faster execution and clearer simulator definition (Lirov et al., 1988). For this kind of approach a series of hierarchical dynamic goal-oriented systems should be developed and investigated.

There are many such problems where human expert skills in reasoning about complex goal-oriented systems are incomparably higher than the level of modern computing systems. Unfortunately, problems of tactics planning and automatic control of autonomous agents such as aerospace vehicles, space stations and robots with cooperative and opposing interests are of the type where human problem-solving skills can not be directly applied. Moreover, there are no highly-skilled human experts in these fields ready to substitute for robots (on a virtual model) or transfer their knowledge to them. There is no grand-master in robot control, although, of course, the knowledge of existing experts in this field should not be neglected – it is even more valuable. Due to the special significance of these problems and the fabulous costs of mistakes, the quality of solutions must be very high and usually subject to continuous improvement.

In this respect it is very important to study human expert reasoning about similar complex systems in the areas where the results are successful, in order to discover the keys to success, and then apply and adopt these keys to the new, as yet, unsolved problems, and first and foremost to the aerospace critical complex systems. It should be considered as investigation, development, and consequent expansion of advanced human expert skills into new areas.

The question that remains then, is this: what language tools do we have for the adequate representation of human expert skills? An application of such language tools to an area of successful results achieved by a human expert should yield a *formal, domain independent knowledge* ready to be transferred to different areas. Neither natural nor programming languages satisfy our goal. The first are informal and ambiguous, while the second are usually detailed,

lower-level tools. Actually, we have to learn how we can formally represent, generate, and investigate a *mathematical model* based on the *abstract images* extracted from the expert vision of the problem.

2. BACKGROUND

The difficulties we encounter trying to find the optimal operation for real-world complex control systems are well known. While the formalization of the problem, as a rule, is not difficult, an algorithm that finds its solution usually results in the search of many variations. For small-dimensional "toy" problems a solution can be obtained; however, for most real-world problems the dimension increases and the number of variations increases significantly, usually exponentially, as a function of dimension (Garey and Johnson, 1991). Thus, most real-world search problems are not solvable with the help of exact algorithms in a reasonable amount of time. This becomes increasingly critical for the real-time aerospace autonomous and semiautonomous vehicles and robots (Lirov et al., 1988; Strosnider and Paul, 1994).

There have been many attempts to find the optimal (suboptimal) operation for real-world complex systems, in particular, for aerospace applications (Leitmann, 1990; Drabble, 1991; Pigeon et al., 1992). Basically, all the approaches for the limited time search can be broken into four categories: the imprecise computation (Chung et al., 1990), real-time search (e.g., Korf, 1990), approximate processing (Lesser et al., 1988), and anytime algorithms (Dean and Boddy, 1988). According to Strosnider and Paul (1994) the correct pruning in its many manifestations is still the only technique that reduces the worst-case execution time without compromising the goal state. But for real-world applications this reduction is usually insufficient: it does not overcome the combinatorial explosion. Another techniques, such as approximate processing, scoping, and use of domain knowledge, can reduce execution time significantly but they might compromise the goal state.

One of the basic ideas is to decrease the dimension of the real-world system following the approach of a *human expert in the field*, by breaking the system into smaller subsystems. This process of decomposition can be applied recursively until we end up with a collection of basic subproblems that can be treated (in some sense) independently. These ideas have been

implemented for many problems with varying degrees of success (see, e.g., Albus, 1991; Knoblock, 1990; Mesarovich et al., 1989; Botvinnik, 1984). Implementations based on the formal theories of linear and nonlinear planning meet hard efficiency problems (McAllester and Rosenblitt, 1991; Chapman, 1987; Nilsson, 1980; Stefik, 1981; Sacerdoti, 1975). An efficient planner requires an intensive use of heuristic knowledge. Moreover, it is possible to use both dynamic and static heuristic knowledge in reducing the search variations. The dynamic knowledge can be acquired during the run time and immediately applied for search reduction (Strosnider and Paul, 1994). On the other hand, a pure heuristic implementation is unique. There is no general constructive approach to such implementations. Each new problem should be carefully studied, and previous experience usually can not be applied. Basically, we can not answer the question: what are the formal properties of the human expert heuristics that drove us to a successful hierarchy of subsystems for a given problem, and how can we apply the same ideas in a an altered or even different problem domain? Moreover, every attempt to evaluate the computational complexity and quality of a pilot solution necessitates implementing its program, which in itself is a unique task for each problem.

In the 1960's, a formal syntactic approach to the investigation of properties of natural language resulted in the fast development of a theory of formal languages by Chomsky (1963), Ginsburg (1966), and others. This development provided an interesting opportunity for dissemination of this approach to different areas. In particular, there came an idea of analogous linguistic representation of images. This idea was successfully developed into syntactic methods of pattern recognition by Fu (1982), Narasimhan (1966), and Pavlidis (1977), and picture description languages by Shaw (1969), Feder (1971), and Rosenfeld (1979).

Searching for adequate mathematical tools formalizing human heuristics of dynamic hierarchies, we have transformed the idea of linguistic representation of complex real-world and artificial images into the idea of similar representation of complex hierarchical systems (Stilman, 1985). However, the appropriate languages should possess more sophisticated attributes than languages usually used for pattern description. The origin of such languages can be traced back to the research on programmed

attribute grammars by Knuth (1968), Rozenkrantz (1969).

A mathematical environment (a “glue”) for the formal implementation of this approach was developed following the theories of formal problem solving and planning by Nilsson (1980), Fikes and Nilsson (1971), Sacerdoti (1975), McCarthy (1980), McCarthy and Hayes (1969), and others based on first order predicate calculus.

In the beginning of 80’s Botvinnik, Stilman, and others developed one of the most interesting and powerful heuristic hierarchical models. It was successfully applied to scheduling, planning, control, and computer chess. The hierarchical networks were introduced in (Botvinnik, 1984; Stilman, 1977) in the form of ideas, plausible discussions, and program implementations (see below). We consider this model as an ideal case for transferring the developed search heuristics to the aerospace domain employing formal linguistic tools.

An application of the developed model to a chess domain was implemented in full as program PIONEER (Botvinnik, 1984). Similar heuristic model was implemented for power equipment maintenance in a number of computer programs being used for maintenance scheduling all over the USSR (Botvinnik et al., 1983; Reznitskiy and Stilman, 1983; Stilman, 1985, 1993a). All these earlier developed programs were the direct implementations of the specific dynamic hierarchies of subsystems. The first pilot implementation of the elements of the generic hierarchy of formal languages for the 2D case was done at the University of Colorado at Denver by King (1993) and Mathews (1993) employing CLIPS tools (Giarratano, 1991) and C language, respectively.

The results shown by these programs in solving complex chess and scheduling problems indicate that implementations of the dynamic hierarchy resulted in the extremely goal-driven algorithms generating search trees with a branching factor close to 1.

In order to discover the inner properties of human expert heuristics, which have been successful in a certain class of complex control systems, we develop a formal theory, the so-called *Linguistic Geometry* (Stilman, 1993-94). This research includes the development of syntactic tools for *knowledge representation* and *reasoning* about large-scale hierarchical complex systems. It relies on the formalization of *search heuristics*, which allow one to decompose complex system

into a hierarchy of subsystems, and thus solve intractable problems by reducing the search. These *hierarchical images* in the form of networks of paths were extracted from the expert vision of the problem. The hierarchy of subsystems is represented as a *hierarchy of formal attribute languages*.

3. SHORT SURVEY OF LINGUISTIC GEOMETRY

In order to pursue our objectives formally we have to define a class of problems to be studied and introduce a hierarchy of languages for decomposition of these problems.

3.1 Class of Problems

A Complex System is the following eight-tuple:

$\langle X, P, R_p, \{ON\}, v, S_i, S_t, TR \rangle$, where

$X = \{x_j\}$ is a finite set of *points*;

$P = \{p_i\}$ is a finite set of *elements*; P is a union of two non-intersecting subsets P_1 and P_2 ;

$R_p(x, y)$ is a set of binary relations of *reachability* in X (x and y are from X , p from P);

$ON(p)=x$, where ON is a partial function of *placement* from P into X ;

v is a function on P with positive integer values describing the *values* of elements.

The Complex System searches the state space, which should have initial and target states;

S_i and S_t are the descriptions of the *initial* and *target* states in the language of the first order predicate calculus, which matches with each relation a certain Well-Formed Formula (WFF). Thus, each state from S_i or S_t is described by a certain set of WFF of the form $\{ON(p_j) = x_k\}$;

TR is a set of operators, $TRANSITION(p, x, y)$, of transitions of the System from one state to another one. These operators describe the transition in terms of two lists of WFF (to be removed from and added to the description of the state), and of WFF of applicability of the transition. Here,

Remove list: $ON(p)=x, ON(q)=y$;

Add list: $ON(p)=y$;

Applicability list: $(ON(p)=x)^{\wedge}R_p(x,y)$, where p belongs to P_1 and q belongs to P_2 or vice versa. The transitions are carried out with participation of one or many elements p

from P_1 and P_2 .

According to the definition of the set P , the elements of the System are divided into two subsets P_1 and P_2 . They might be considered as units moving along the reachable points. Element p can move from point x to point y if these points are reachable, i.e., $R_p(x, y)$ holds. The current location of each element is described by the equation $ON(p)=x$. Thus, the description of each state of the System $\{ON(p_j) = x_k\}$ is the set of descriptions of the locations of the elements. The operator $TRANSITION(p, x, y)$ describes the change of the state of the System caused by the move of the element p from point x to point y . The element q from point y must be withdrawn (eliminated) if p and q do not belong to the same one of the two subsets P_1 and P_2 .

The problem of the optimal operation of the System is considered as a search for the optimal sequence of transitions leading from one of the initial states of S_i to a target state S of S_t .

It is easy to show formally that a robotic system can be considered as a Complex System (see below). Many different technical and human society systems (including military battlefield systems, systems of economic competition, positional games) that can be represented as twin sets of movable units (representing two or more opposing sides) and their locations can be considered as Complex Systems.

With such a problem statement for the search of the optimal sequence of transitions leading to the target state, we could use formal methods like those in the problem-solving system STRIPS (Fikes and Nilsson, 1971), nonlinear planner NOAH (Sacerdoti, 1975), or in subsequent planning systems. However, the search would have to be made in a space of a huge dimension (for nontrivial examples). Thus, in practice no solution would be obtained.

We devote ourselves to finding an approximate solution of a reformulated problem.

3.2 Geometry of Complex Systems: Measurement of Distances

To create and study a hierarchy of dynamic subsystems, we have to investigate geometrical properties of the Complex System.

A *map of the set X* relative to the point x and element p for the Complex System is the mapping: $MAP_{x,p}: X \rightarrow \mathbb{Z}_+$ (where x is from X , p is from P), which is constructed as follows.

We consider a *family of reachability areas* from the point x , i.e., a finite set of the following nonempty subsets $\{M^k_{x,p}\}$ of X (Fig.1):

$k=1$: $M^k_{x,p}$ is a set of points m *reachable in one step* from x : $R_p(x,m)=T$;

$k>1$: $M^k_{x,p}$ is a set of points *reachable in k steps and not reachable in k-1 steps*, i.e., points m reachable from points of $M^{k-1}_{x,p}$ and not included in any $M^i_{x,p}$ with i less than k .

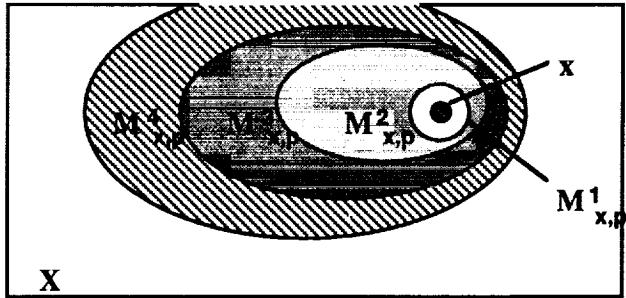


Fig. 1. Interpretation of the family of reachability areas
Let $MAP_{x,p}(y)=k$, for y from $M^k_{x,p}$ (*the number of steps from x to y*). For the remaining points, let $MAP_{x,p}(y)=2n$, if $y \neq x$ (n is the number of points in X); $MAP_{x,p}(y)=0$, if $y = x$.

It is easy to verify that the map of the set X for the specified element p from P defines an *asymmetric distance function* on X :

1. $MAP_{x,p}(y) > 0$ for $x \neq y$; $MAP_{x,p}(x)=0$;
2. $MAP_{x,p}(y)+MAP_{y,p}(z) \geq MAP_{x,p}(z)$.

If R_p is a symmetric relation,

3. $MAP_{x,p}(y)=MAP_{y,p}(x)$.

In this case each of the elements p from P specifies on X its *own metric*.

Various examples of measurement of distances for robotic vehicles are considered in (Stilman, 1993a, 1993c).

3.3 Set of Paths: Language of Trajectories

This language is a formal description of the set of lowest-level subsystems, the set of all paths between points of the Complex System. An element might follow a path to achieve the goal "connected with the ending point" of this path.

A *trajectory* for an element p of P with the beginning at x of X and the end at the y of X ($x \neq y$) with a length l is following formal string of symbols $a(x)$ with points of X as parameters:

$$t_0 = a(x)a(x_1)\dots a(x_l),$$

where $x_l = y$, each successive point x_{i+1} is

reachable from the previous point x_i , i.e., $R_p(x_i, x_{i+1})$ holds for $i = 0, 1, \dots, l-1$; element p stands at the point x : $ON(p)=x$. We denote by $t_p(x, y, l)$ the set of all trajectories for element p , beginning at x , end at y , and with length l . $\mathcal{P}(t_0)=\{x, x_1, \dots, x_l\}$ is the set of parameter values of the trajectory t_0 . (To avoid confusion we should emphasize that $a(x)a(x_1)\dots a(x_l)$ is a formal record and does not mean anything else except what is given above.)

A *shortest trajectory* t of $t_p(x, y, l)$ is the trajectory of minimum length for the given beginning x , end y , and element p .

Properties of the Complex System permit us to define (in general form) and study formal grammars for generating the shortest trajectories. A general grammar and its application to generating the shortest trajectory for a aerospace robotic vehicle will be presented later.

Reasoning informally, an analogy can be set up: the shortest trajectory is analogous with a straight line segment connecting two points in a plane. An analogy to a k -element segmented line connecting these points is called an *admissible trajectory of degree k* , i.e., the trajectory that can be divided into k shortest trajectories. The admissible trajectories of degree 2 play a special role in many problems. As a rule, elements of the System should move along the shortest paths. In case of an obstacle, the element should move around this obstacle by tracing an intermediate point aside and going to and from this point to the end along the shortest trajectories. Thus, in this case, an element should move along an admissible trajectory of degree 2.

A *Language of Trajectories* $L_t^H(S)$ for the Complex System in a state S is the set of all the shortest and admissible (degree 2) trajectories of length less than H . Different properties of this language and generating grammars were investigated in (Stilman, 1993a).

3.4 Networks of Paths: Languages of Trajectory Networks

After defining the Language of Trajectories, we have new tools for the breakdown of our System into subsystems. According to the ideas presented in (Botvinnik, 1984), these subsystems should be various types of trajectory networks, i.e., the sets of interconnected trajectories with one singled out and called the *main trajectory*. An example of such network is shown in Fig. 2. The basic idea behind

these networks is as follows. Element p_0 should move along the main trajectory $a(1)a(2)a(3)a(4)a(5)$ to reach the ending point 5 and remove the target q_4 (an opposing element). Naturally, the opposing elements should try to disturb those motions by controlling the intermediate points of the main trajectory. They should come closer to these points (to the point 4 in Fig. 2) and remove element p_0 after its arrival (at point 4). For this purpose, elements q_3 or q_2 should move along the trajectories $a(6)a(7)a(4)$ and $a(8)a(9)a(4)$, respectively, and wait (if necessary) on the next to last point (7 or 9) for the arrival of element p_0 at point 4. Similarly, element p_1 of the same side as p_0 might try to disturb the motion of q_2 by controlling point 9 along the trajectory $a(13)a(9)$. It makes sense for the opposing side to include the trajectory $a(11)a(12)a(9)$ of element q_1 to prevent this control.

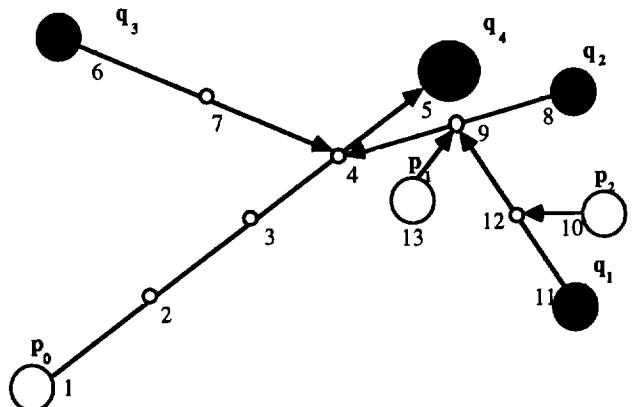


Fig. 2. Network language interpretation.

Similar networks are used for the breakdown of complex systems in different areas. Let us consider a linguistic formalization of such networks. The Language of Trajectories describes "one-dimensional" objects by joining symbols into a string employing a reachability relation $R_p(x, y)$. To describe networks, i.e., "multi-dimensional" objects made up of trajectories, we use the relation of *trajectory connection*.

A *trajectory connection* of the trajectories t_1 and t_2 is the relation $C(t_1, t_2)$. It holds if the ending link of the trajectory t_1 coincides with an intermediate link of the trajectory t_2 ; more precisely, t_1 is connected with t_2 if among the parameter values $\mathcal{P}(t_2)=\{y, y_1, \dots, y_l\}$ of trajectory t_2 there is a value $y_i = x_k$, where

$t_1=a(x_0)a(x_1)\dots a(x_k)$. If t_1 belongs to a set of trajectories with the common end-point, then the entire set is said to be connected with the trajectory t_2 .

For example, in Fig. 2 the trajectories $a(6)a(7)a(4)$ and $a(8)a(9)a(4)$ are connected with the main trajectory $a(1)a(2)a(3)a(4)a(5)$ through point 4. Trajectories $a(13)a(9)$ and $a(11)a(12)a(9)$ are connected with $a(8)a(9)a(4)$.

To formalize the trajectory networks, we define and use routine operations on the set of trajectories: $C_A^k(t_1, t_2)$, a *k-th degree of connection*, and $C_A^+(t_1, t_2)$, a *transitive closure*.

Trajectory $a(11)a(12)a(9)$ in Fig. 2 is connected degree 2 with trajectory $a(1)a(2)a(3)a(4)a(5)$, i.e., $C^2(a(11)a(12)a(9), a(1)a(2)a(3)a(4)a(5))$ holds. Trajectory $a(10)a(12)$ in Fig. 2 is in transitive closure to the trajectory $a(1)a(2)a(3)a(4)a(5)$ because $C^3(a(10)a(12), a(1)a(2)a(3)a(4)a(5))$ holds by means of the chain of trajectories $a(11)a(12)a(9)$ and $a(8)a(9)a(4)$.

A *trajectory network* W relative to trajectory t_0 is a finite set of trajectories t_0, t_1, \dots, t_k from the language $L_t^H(S)$ that possesses the following property: for every trajectory t_i from W ($i = 1, 2, \dots, k$) the relation $C_W^+(t_i, t_0)$ holds, i.e., each trajectory of the network W is connected with the trajectory t_0 that was singled out by a subset of interconnected trajectories of this network. If the relation $C_W^m(t_i, t_0)$ holds, i.e., this is the m -th degree of connection, trajectory t_i is called the m *negation trajectory*.

Obviously, the trajectories in Fig. 2 form a trajectory network relative to the main trajectory $a(1)a(2)a(3)a(4)a(5)$. We are now ready to define network languages.

A *family of trajectory network languages* $L_C(S)$ in a state S of the Complex System is the family of languages that contains strings of the form

$$t(t_1, param)t(t_2, param)\dots t(t_m, param),$$

where *param* in parentheses substitute for the other parameters of a particular language. All the symbols of the string t_1, t_2, \dots, t_m correspond to trajectories that form a trajectory network W relative to t_1 .

Different members of this family correspond to different types of trajectory network languages,

which describe particular subsystems for solving search problems. One such language is the language that describes specific networks called Zones. They play the main role in the model considered here (Botvinnik, 1984; Stilman, 1977, 1993b, 1993c, 1994a). A formal definition of this language is essentially constructive and requires showing explicitly a method for generating this language, i.e., a certain formal grammar, which is presented in (Stilman, 1993b, 1993c, 1994a). In order to make our points transparent here, we define the Language of Zones informally.

A *Language of Zones* is a trajectory network language with strings of the form

$$Z=t(p_0, t_0, \tau_0) t(p_1, t_1, \tau_1)\dots t(p_k, t_k, \tau_k),$$

where t_0, t_1, \dots, t_k are the trajectories of elements

p_0, p_1, \dots, p_k respectively; $\tau_0, \tau_1, \dots, \tau_k$ are nonnegative integers that “denote the time allotted for the motion along the trajectories” in a correspondence to the mutual goal of this Zone: to remove the target element – for one side, and to protect it – for the opposing side. Trajectory $t(p_0, t_0, \tau_0)$ is called the *main trajectory* of the Zone. The element q standing on the ending point of the main trajectory is called the *target*. The elements p_0 and q belong to the opposing sides.

To make it clearer, let us show the Zone corresponding to the trajectory network in Fig. 2.

$$Z=t(p_0, a(1)a(2)a(3)a(4)a(5), 4)$$

$$t(q_3, a(6)a(7)a(4), 3)$$

$$t(q_2, a(8)a(9)a(4), 3) t(p_1, a(13)a(9), 1)$$

$$t(q_1, a(11)a(12)a(9), 2) t(p_2, a(10)a(12), 1)$$

Assume that the goal of the white side is to remove target q_4 , while the goal of the black side is to protect it. According to these goals, element p_0 starts the motion to the target, while black starts in its turn to move elements q_2 or q_3 to intercept element p_0 . Actually, only those black trajectories are to be included into the Zone where the motion of the element makes sense, i.e., the *length of the trajectory is less than the amount of time (third parameter τ) allocated to it*. For example, the motion along the trajectories $a(6)a(7)a(4)$ and $a(8)a(9)a(4)$ makes sense, because they are of length 2 and time allocated equals 3: each of the elements has 3 time intervals to reach point 4 to intercept element p_0 assuming one would go along the main trajectory without move omission. According to definition of Zone, the trajectories of white elements (except p_0) could only be of the

length 1, e.g., $a(13)a(9)$ or $a(10)a(12)$. As element p_1 can intercept the motion of the element q_2 at the point 9, black includes into the Zone the trajectory $a(11)a(12)a(9)$ of the element q_1 , which has enough time for motion to prevent this interception. The total amount of time allocated to the whole group of black trajectories connected (directly or indirectly) with the given point of the main trajectory is determined by the number of that point. For example, for the point 4, it equals 3 time intervals.

Besides Zones considered above we introduce *retreat* and *unblock* Zones. They include a target (or blocking element) with all possible trajectories of the length 1 with the beginning at the location of this element.

A language $L_Z^H(S)$ generated by the certain grammar G_Z (Stilman, 1993b, 1993c, 1994a) in a state S of a Complex System is called the *Language of Zones*.

Network languages allow us to describe the "statics", i.e., the states of the System. In order to describe the "dynamics" of the System, i.e., the motions from one state to another, we have to regenerate the entire hierarchy of languages. Of course, it is an inefficient procedure. To improve the efficiency of applications in the search process it is important to describe the change of the hierarchy of languages (Stilman, 1994a). A study of this change helped us in modifying the hierarchy instead of regenerating it in each state. This change is represented as a mapping (translation) to some other hierarchy (actually, to the new state of the same hierarchy). Thus, the functioning of the system, in a search process, generates a tree of translations of the hierarchy of languages. This tree is represented as a string of the highest level formal language, the Language of Translations (Stilman, 1994b, 1994c).

A practicality of the formal constructions considered in Section 3 as well as the entire hierarchy of languages are demonstrated on the following 3D example of the problem of strategic planning for the system of simplified space autonomous vehicles.

4. STRATEGIC PLANNING FOR SPACE COMBAT

The combat robotic model can be represented as a Complex System naturally (Section 3.1). The set X represents the operational district, which could be the area of combat operation, broken into smaller square or cubic areas, "points", e.g., in

the form of the big square or cubic grid. It could be a space operation, where X represents the set of different orbits, or an air force battlefield, etc. P is the set of robots or autonomous vehicles. It is broken into two subsets P_1 and P_2 with opposing interests; $R_p(x,y)$ represent moving capabilities of different robots for different problem domains: robot p can move from point x to point y if $R_p(x,y)$ holds. Some of the robots can crawl, others can jump or ride, sail and fly, or even move from one orbit to another. Some of them move fast and can reach point y (from x) in "one step", i.e., $R_p(x,y)$ holds, others can do that in k steps only, and many of them can not reach this point at all. $ON(p)=x$, if robot p is at the point x ; $v(p)$ is the value of robot p . This value might be determined by the technical parameters of the robot. It can include the immediate value of this robot for the given combat operation; S_i is an arbitrary initial state of operation for analysis, or the starting state; S_t is the set of target states. These might be the states where robots of each side reached specified points. On the other hand, S_t can specify states where opposing robots of the highest value are destroyed. The set of WFF $\{ON(p_j) = x_k\}$ corresponds to the list of robots with their coordinates in each state. $TRANSITION(p, x, y)$ represents the move of the robot p from the location x to location y ; if a robot of the opposing side stands on y , a removal occurs, i.e., robot on y is destroyed and removed.

4.1. Problem Statement

Space robotic vehicles with different moving capabilities are shown in Fig. 3. The operational district X is the space grid of $8 \times 8 \times 8$. The total number of cubic areas $n = 512$. Robot W-CENTER (White Command & Control Space Center) located at 818 ($x = 8, y = 1, z = 8$), can move to any next location within the current orbital plane $x1z$, e.g., from its' current location — to 817, 717, 718. Robot B-CENTER (Black Command & Control Space Center) located at 615, can move to any next area within the same plane $x1z$ similarly to the robot W-CENTER. Two other vehicles W-CARRIERS (White Space Buster Carriers) from 715 and 815, respectively, can move only straight ahead towards the strategic goal areas 718 and 818, one square at a time, e.g., from 715 to 716, from 716 to 717, etc. Basically, any of the cubes with the coordinates $y = 1, z = 8$ is desirable for these CARRIERS. Each of the

CARRIERS carries on the top an advanced W-AS-FIGHTER (White Space Fighter) which can take off from the CARRIER only in the strategic districts considered above. After take-off W-AS-FIGHTER can move in any direction, diagonally or straight forward or backward moving through several cubic areas during one time unit. The B-CARRIER at 351 is analogous to W-CARRIERS. It can move only straight ahead toward the strategic goal area 311 where the B-AS-FIGHTER, the cargo, can take off.

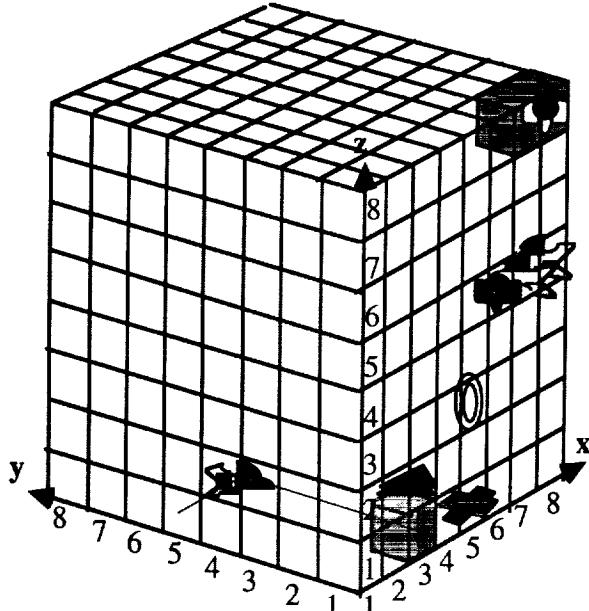


Fig. 3. A problem for space robotic vehicles.
The vehicle W-STATION (White Space Station) located at 513 can move only straight ahead one cubic area at a time, i.e., from 513 to 514, from 514 to 515; its mobility is limited by two areas, 514 and 515 only. The rest of Black vehicles are B-INTERCEPTOR (Black Interceptor) and B-SCOUT (Black Scout-Fighter). B-INTERCEPTOR located at 312 can move diagonally with several cubic areas at a time, e.g., from 312 to 114 or to 514. Finally, B-SCOUT looking for a strategic information can leap forward, backward or right or left two squares at a time, e.g., from 511 it can move to 712, 613, 413. All these vehicles can move only within the current orbital plane $x1z$.

Theoretically, B-SCOUT at 511 can reach any of the points $z \in \{712, 613, 413, 312, 721\}$ in one step, i.e., $R_{B-SCOUT}(511, z)$ holds, while B-INTERCEPTOR can reach $z \in \{211, 413, 514, 615, 716, 817, 411, 213, 114\}$ in one step, i.e., $R_{B-INTERCEPTOR}(312, z)$ holds. Assume that

the grid is so fine that none of the vehicles can move through the cubic area where another vehicle is currently located (or stop in this area). This means that in the current state B-INTERCEPTOR actually can not move to 615, 716, 817, while B-SCOUT can not leap to 312.

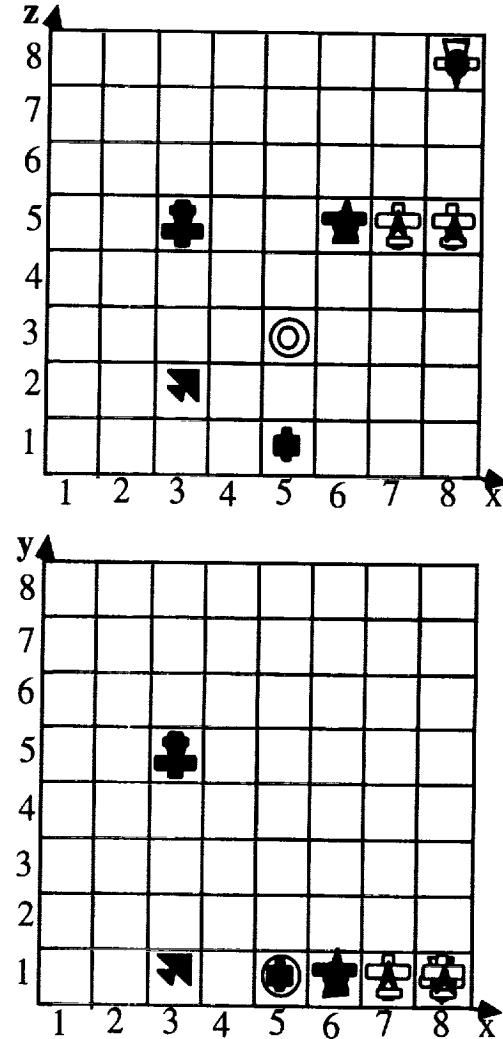


Fig. 3. A problem for space vehicles (2 projections).
Assume also that vehicles W-CENTER, W-STATION, and W-CARRIERS including, of course, their cargo, W-AS-FIGHTERS, belong to one side, while B-CENTER, B-INTERCEPTOR, B-SCOUT, and B-CARRIER with its cargo belong to the opposing side. This means that these agents belong to the sets P_1 and P_2 , respectively. Each of the vehicles has powerful weapons able to destroy opposing vehicles ahead of the course, and this way move through the area where this vehicle is currently located. For example, B-CENTER from 615 can move to 815 through 715 in two steps destroying both W-CARRIERS along

the way. The only difference is with the White and Black CARRIERS and W-STATION. While routinely they can move only straight ahead (and be blocked by any of the friendly or opposing vehicles), they can destroy opposing vehicles at the next diagonal locations ahead of the course and then move to their respective areas. For example, W-CARRIER from 715 can destroy opposing B-CENTER at 616 and 816 and move to its' location. In particular, this diagonal attack ability extends the mobility of W-STATION to the areas 616 and 416 where it can hit the opposing side vehicle. Obviously, each of the opposing sides must avoid loosing a respective W(B)-CENTER which means a complete destruction of the command and control battlefield communications and immediately ends the combat in a loss to this side. On the other hand, launching a totally powerful Aerospace Fighter (AS-FIGHTER) and preventing lunch of the opposing AS-FIGHTER (or destroying it) is considered as a win. The conditions considered above give us S_t , the description of target states of the Complex System. The description of the initial state S_i is obvious and follows from Fig. 3 (xz- and yz-projections).

Assume that our time scale discretization is such that motions of the opposing sides alternate, and due to the shortage of resources (which is typical in a real combat operation) or some other reasons, each side can not participate in two missions simultaneously. It means that during the current time unit, in case of the White turn, only one of the White vehicles can move. Analogous condition holds for Black. Of course, it does not mean that if one side began participating in one of the missions, it must complete it. Any time on its turn each side can switch from one mission to another, e.g., transferring resources (fuel, weapons, human resources, etc.), and later switch back.

Similar to the real world operation it is hard to predict the result of this simplified combat. However, it seems that the locations of the W-CARRIERS are advantageous in comparison with the Black agents, B-CENTER, B-INTERCEPTOR, and B-SCOUT, while B-CARRIER is too far from the strategic area at 311. It is likely that Black can not prevent lunches of W-AS-FIGHTERS (or destroy both of them). Is there a strategy for the Black side to win or, at least, end this combat in a draw lunching its B-AS-FIGHTER on time?

Of course, this question can be answered by a direct search employing, for example, the minimax algorithm with alpha-beta cut-offs. Theoretical evaluations and experiments with computer showed that finding a solution of this problem requires generation of the search tree that includes about 30^{25} moves (transitions). Of course, this is beyond reasonable time constraints of the most advanced modern computers. It is very interesting to observe the dramatic reduction of search employing the Linguistic Geometry tools.

In order to demonstrate generation of the Hierarchy of Languages for this problem, we have to generate the Language of Trajectories and the Language of Zones in each state of the search. The details of generation of trajectories and Zones for 2D and 3D problems are considered in (Stilman, 1993b, 1993c, 1993d, 1994b, 1994c).

4.2. Search Generation for Space Combat

Consider how the hierarchy of languages works for the optimal control of the Space Combat System introduced above (Fig. 3). We generate the string of the Language of Translations (Stilman, 1994a) representing it as a conventional search tree (Fig.1) and comment on its generation.

In our comments on this generation we will emphasize the major steps and skip some technical details considered, e.g., in (Stilman, 1994c).

First, the Language of Zones in the start state is generated. Every element tries to attack every element of the opposing side. The targets for attack are determined within the limit of four steps, the horizon. This is a "view range" of this problem. It means that horizon H of the language $L_Z(S)$ is equal to 4, i.e., the length of the main trajectories of all Zones must not exceed 4 steps. The reasons and the algorithm for choosing the right value of the horizon are considered in (Stilman, 1994c). One of the Zones for W-CARRIER at 715, Z_{WC} is shown in Fig.1. In formal notation this Zone is as follows:

$Z_{WC} =$

```
t(W-CARRIER, a(715)a(716)a(717)a(718), 4)
t(B-CENTER, a(615)a(716), 2)
t(B-CENTER, a(615)a(616)a(717), 3)
t(B-CENTER, a(615)a(616)a(617)a(718), 4)
t(B-INTERCEPTOR, a(312)a(817)a(718), 4)
t(W-CARRIER, a(815)a(715), 1)
t(W-CENTER, a(818)a(718), 1)
t(W-CENTER, a(818)a(717), 1)
t(W-CENTER, a(818)a(817), 1)
```

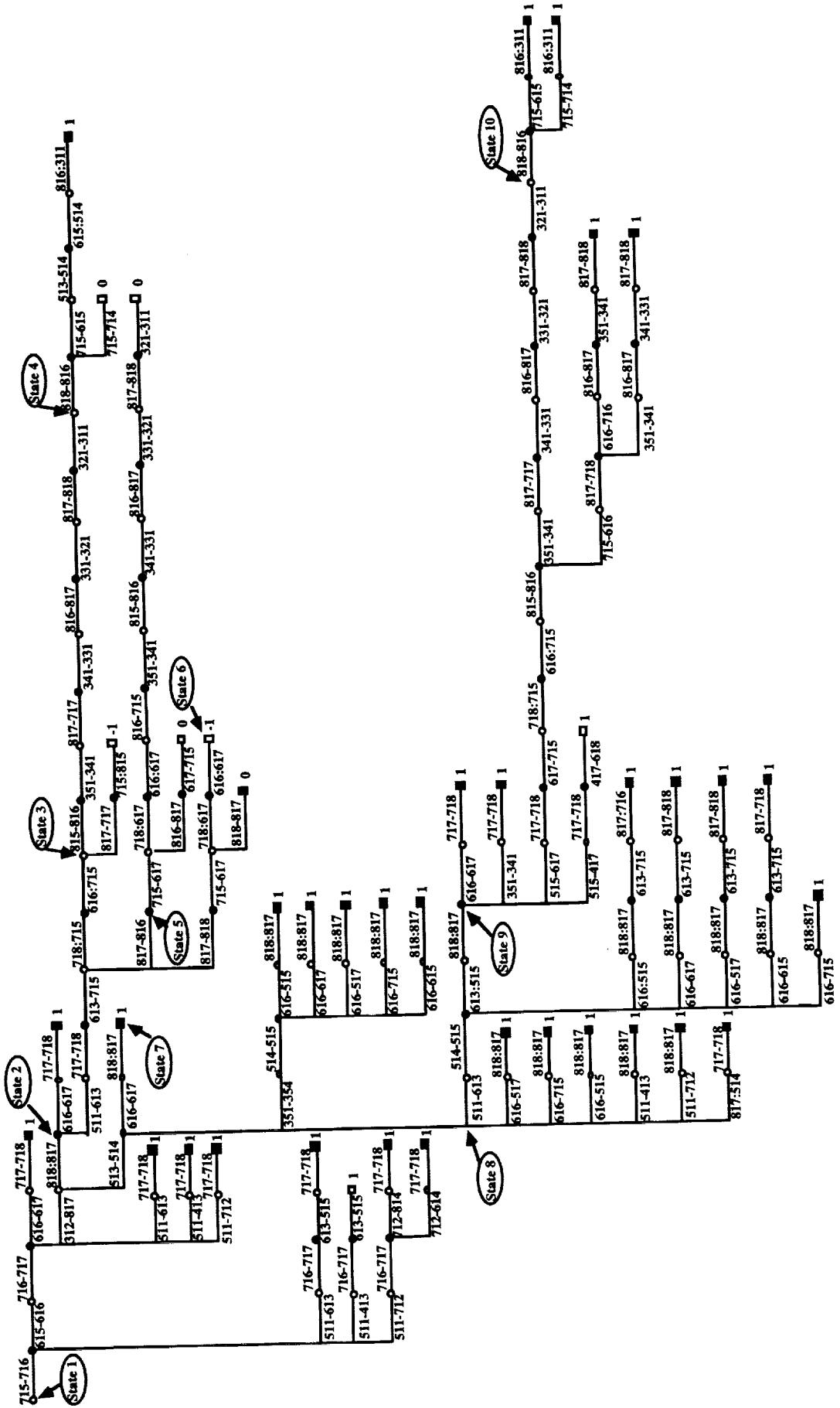


Fig. 4. Search tree for the optimization problem for space vehicles within the horizon 4.

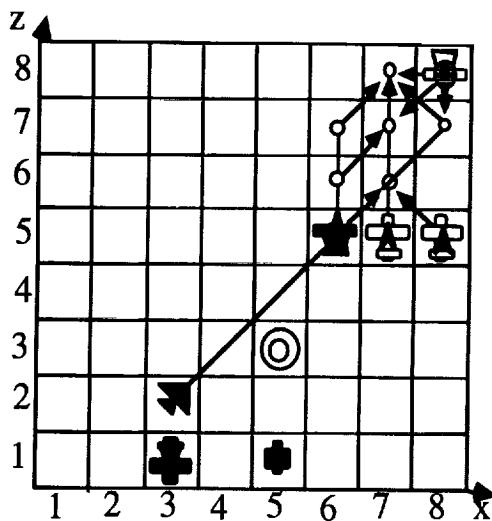


Fig. 5. State 1

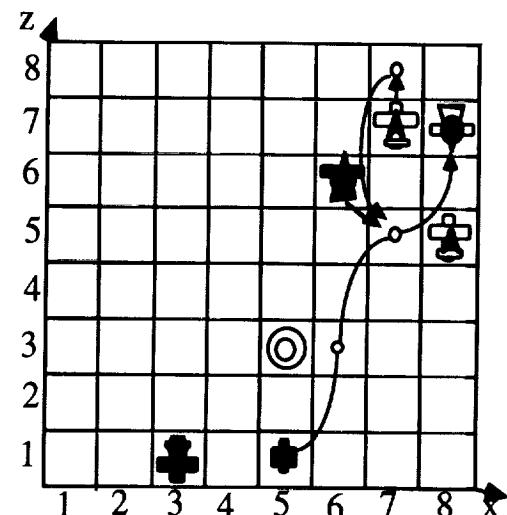


Fig. 6. State 2

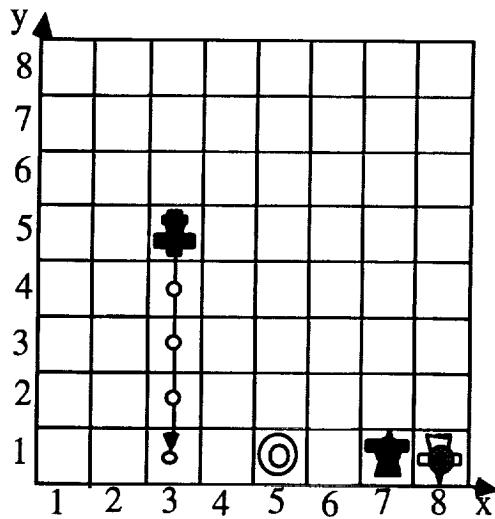


Fig. 7. State 3 (xy- and xz-projections)

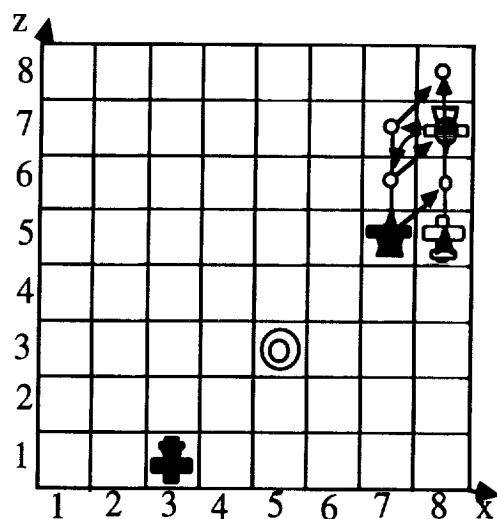


Fig. 8. State 4

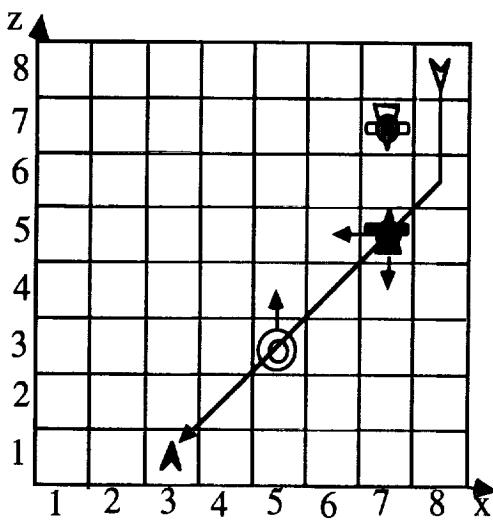


Fig. 9. State 5

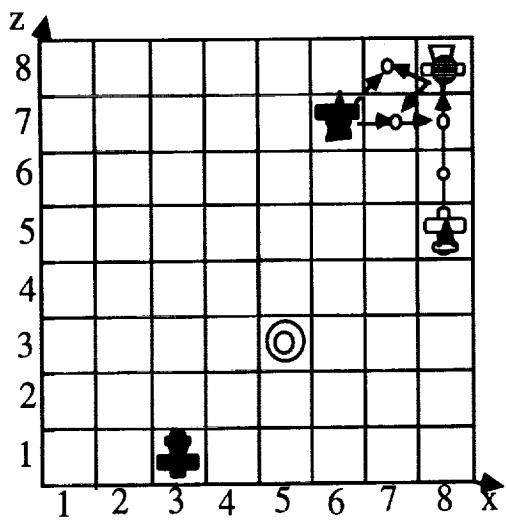


Fig. 10. State 6

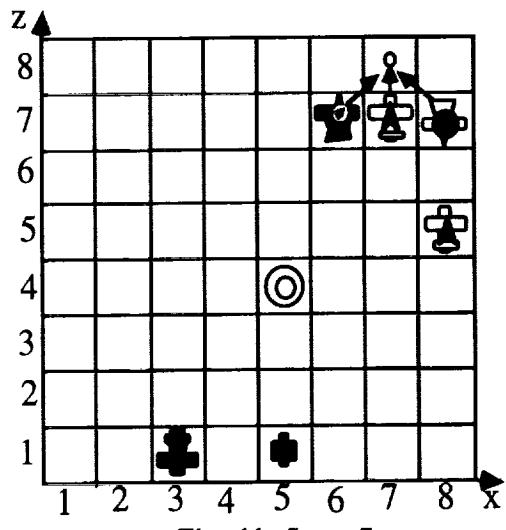


Fig. 11. State 7

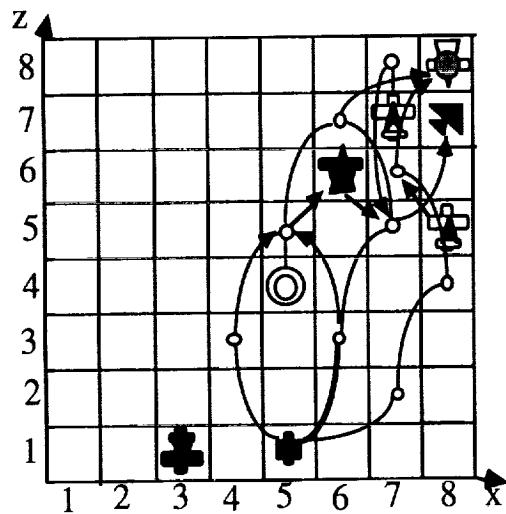


Fig. 12. State 8

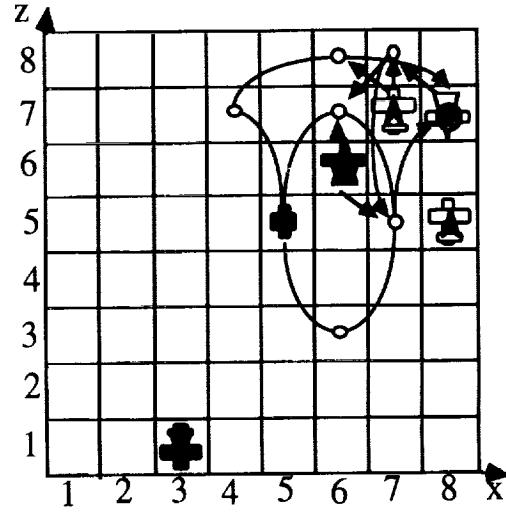


Fig. 13. State 9

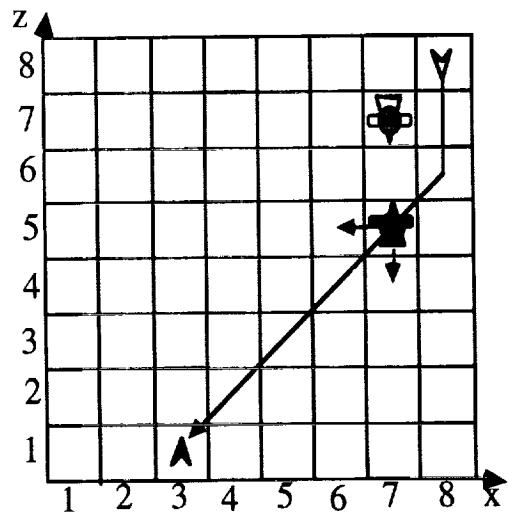


Fig. 14. State 10

Search tree generation (Fig. 4) begins with the move 1. 715-716 in the most traversable White Zone with the vulnerable target of the highest value. This Zone Z_{WC} of W-CARRIER is shown in Fig. 5 (yz-projection). The order of consideration of Zones and particular trajectories is determined by the Grammar of Translations. The computation of move-ordering constraints is the most sophisticated procedure in the Grammar of Translations. It takes into account different parameters of Zones, trajectories, and the so-called chains of trajectories. We should keep in mind that after each move the model moves to the new current state S_c, so the entire Language of Zones, L_Z(S_c), must be regenerated. With respect to efficiency of the model it was very important to solve a technical problem relative to the well known Frame Problem (McCarthy and Hayes, 1969; Fikes and Nilsson, 1971; Nilsson, 1980). This allowed us to avoid recomputation of the entire language recomputing only the changing part. An approach to the formal solution of this problem is considered in (Stilman, 1994a).

The next move, 1. ... 615-616, is in the same Zone along the first negation trajectory. B-CENTER is trying to intercept motion of the W-CARRIER at 717 or 718. The interception continues: 2. 716-717 616-617 3. 717-718. Interception failed and here the grammar terminates this branch with the value of 1 (as a win of the White side). This value is given by the special state evaluation procedure built into the grammar. This procedure evaluated this state as a winning state for the White after analysis of the "traversability" of all the Zones active in this state. In particular, it figured out that the exchange at 718: 3. ... 617:718 4. 818:718 would destroy B-CENTER and, thus, it is unacceptable for Black. (Here and in the search tree symbol ":" means the removal of an element.) Moreover, the safe arrival of W-CARRIER at the strategic area 718 would cause the lunch of W-AS-FIGHTER ending the combat in a win for the White side. Also, the analysis of the Black Zones showed that Black have nothing to oppose.

The grammar initiates the backtracking climb. After the climb up to the move 2. ... 616-617 different intercepting trajectory in the same Zone (Fig. 5) has been activated $a(312)a(817)a(718)$: 2. ... 312-817. After the arrival at 817 B-FIGHTER has been destroyed by W-CENTER, and the following interception failed: 3. 818:817 616-617 4. 717-718.

The backtracking climb up to the move 3.

818:817 is interrupted at the State 2 shown in Fig. 6. This is the state where the new attacking Zone of B-SCOUT from 511 to 817 has been registered when we visited this state earlier during descent. This information has been stored to be brought to the upper levels of the search tree; the grammar stores these newly generated Zones as idle for possible activation in different states. Each backtracking move is followed by the inspection procedure, the analysis of the subtree generated in the process of the earlier search. After the climb up to the State 2 (Fig. 6), the tree to be analyzed consists of the only branch: 3. ... 616-617 4. 717-718. The inspection procedure determined that the current minimax value (+1) can be "improved" (in favor of the Black side) by destroying the new target at 817, the W-CENTER. This target was staying at 817 in the analyzed subtree. The improvement can be achieved by participation of W-SCOUT from 511, i.e., by inclusion of the currently idle attack Zone with the main trajectory from 511 to 817 (Fig. 6).

The motion of B-SCOUT along the main trajectory $a(511)a(613)a(715)a(817)$ is accompanied by the motion of intercepting element, initially as W-CARRIER, then from 718 as W-AS-FIGHTER 3. ... 511-613 4. 717-718 613-715 5. 718:715 616:715. Thus, W-SCOUT is intercepted but the newly lunched W-AS-FIGHTER is destroyed also. The current state, State 3, is shown in Fig. 7. In this state the state evaluation procedure could not generate a definite value in favor of either side because two attack Zones for W-CARRIER at 815 and B-CARRIER at 351 are traversable (Fig. 7). Both Zones are activated: 6. 815-816 351-341.

Now the unblock Zone of W-CENTER should be activated in order to free the motion of W-CARRIER through 817. The exact location for the unblock, 7. 817-717, is chosen in order to keep protected the most of the squares of the main trajectory: 816, 817, and 818. The race of CARRIERS continues: 7. ... 341-331 8. 816-817 331-321 9. 817-818 321-311. Both White and Black AS-FIGHTERS are ready be lunched, and the state evaluation procedure still can not terminate the branch. The current state, State 4, is shown in Fig. 8.

Among different attack Zones for W-AS-FIGHTER the Zone with the main trajectory $a(818)a(816)a(311)$ is chosen. This is a traversable "time gaining" trajectory attacking two targets simultaneously, B-CENTER at 715 and B-AS-FIGHTER at 311. After 10. 818-816 the

retreat Zone of W-CENTER at 715 is activated. With two possible safe areas for retreat, 714 and 715, the wrong one is chosen first: 10. ... 715-615. New attack Zone of W-STATION $a(513)a(514)a(615)$ is activated immediately because it is the time-gaining unblock trajectory as well: 11. 513-514. This motion of W-STATION actually gained time. W-CENTER has been engaged and it must respond either destroying W-STATION or retreating, and, thus, losing a time interval and passing a move turn to the White side. W-AS-FIGHTER immediately attacks B-AS-FIGHTER along the trajectory just being unblocked: 11. ... 615:514 12. 816:311. The state evaluation procedure terminates the branch and evaluates as +1 in favor of White. The following backtracking climb up to the move 10. 818-616 where the retreat Zone of B-CENTER is activated again. Now the right area for retreat is chosen 10. ... 715-714. In absence of the vulnerable or time-gaining threats from either side the branch is terminated in a draw (0). The "guilty party" for this draw value is W-STATION at 513. The unblock Zone registered in this terminal state as idle is stored to be activated at the upper levels of the search tree.

It seems that our preliminary estimate about easy win of the White side was incorrect. With the precise planning Black forced a draw in the variations analyzed so far. Let us continue the tree generation.

The grammar initiates the backtracking climb up to the State 3 (Fig. 7). Now when we propagate the draw value as an optimum White is changing moves looking for a win. An attempt of the earlier activation of the W-CARRIER unblock Zone fails because White lose the last W-CARRIER with its valuable cargo: 6. 817-717 715:815. The optimum value is still a draw. The climb continues and move 5. 718:715 with B-SCOUT removal (while W-CENTER is under direct threat) is changed for W-CENTER retreat 5. 817-816. The current State 5 is shown in Fig. 9. A new Zone of B-SCOUT with the main trajectory $a(715)a(617)a(816)$ is immediately activated (Fig. 9): 5. ... 715-617 6. 718-617 616:617. B-SCOUT at 617 is intercepted by W-AS-FIGHTER while W-AS-FIGHTER itself is destroyed by B-CENTER. The state evaluation procedure does not generate a definite value in favor of either side and branch generation continues. The following branch is quite similar to the previous long branch which includes the race of W-CARRIER from 815 and B-CARRIER from

351. The difference is that in this variation W-CENTER unblocks the main trajectory from 816 to 715: 7. 816-715, and stays there while B-CENTER is at 617 all the time (compare with Fig. 8). These new locations of White and Black CENTERS result in a draw after the arrival of both CARRIERS at the respective strategic locations, 818 and 311. The state evaluation procedure does not register vulnerable time-gaining threats and terminates this branch.

The grammar initiates the backtracking climb up to the move 5. ... 715-617. In this state the tree inspection procedure activates the W-CENTER retreat Zone from 816 changing B-SCOUT interception 6. 718:617 for the only W-CENTER retreat 6. ... 816-817. The new attack Zone of B-SCOUT with the main trajectory $a(617)a(715)a(817)$ is activated: 7. 617-715. Here the state evaluation procedure registered state repetition in the current branch (compare with the state before State 5 shown in Fig. 9), and terminated the branch with the draw value (0).

The following climb is interrupted in the state after 5. ... 613-715, and W-CENTER retreat move 6. 817-816 is changed for the last possible retreat: 6. 817-818. The new B-SCOUT attack Zone is immediately activated via $a(715)a(617)a(818)$. The intercepting trajectories are similar to the Zone shown in Fig. 9. The following variation 6. ... 715-617 7. 718:617 616:617 is terminated in the state, State 6, shown in Fig. 10. The state evaluation procedure detected that the Zone for W-CARRIER at 815 is non-traversable (because the unblocking of W-CENTER is impossible) while B-CARRIER Zone from 351 is traversable, and evaluated this state as (-1) in favor of Black. The following climb and change of 7. 718:617 for W-CENTER retreat 7. 818-817 results in the state which has already occurred in the search tree and was evaluated as a draw (0).

The backtracking climb continues propagating the value of 0 (a draw) as a minimax value of the currently generated subtree. The climb stops at the move 3. 818:817, which is changed for 3. 513-514. The tree inspection procedure has chosen this move as a move of a very high preference. This is the first time when new Zone of W-STATION at 513 with the main trajectory $a(513)a(514)a(515)a(616)$ is activated. In the backtracking climb B-CENTER returned to 616, and now White could attack this target within the horizon 4. (The actual length of the main trajectory is 3 steps.) Moreover, this is a time-gaining

motion because this is the motion in the unblock Zone of W-STATION. This Zone registered in the bottom of the search tree (Fig. 8), has been idle for a long time, and now is activated as well.

The following motion continues in the Zone of W-CARRIER with participation of the intercepting and protecting elements, B-CENTER and W-CENTER: 3. ... 616-617 4. 818:817. This state is shown in Fig. 11 (State 7). It is evaluated in favor of White (+1), and the branch is terminated. From now on the current minimax value of the subtree generated so far is a win for White. Now Black try to branch. After the climb Black side activates the attack Zone of B-CARRIER at 351, while W-STATION continues attack of B-CENTER: 3. ... 616-617 4. 514-515. In response, Black explore the destruction of the attacker and all possible retreats. In all these cases White continue 5. 818:817 and these branches terminated with the value in favor of White.

After multiple descents and ascents the grammar returns to the State 8 shown in Fig. 12. The tree inspection procedure activates motion of B-SCOUT along the intercepting trajectory $a(511)a(613)a(515)$ (Fig. 12). This trajectory is of high preference because it partly coincides with the main trajectories of two different Zones: $a(511)a(613)a(715)a(617)a(818)$ or $a(511)a(613)a(715)a(617)a(818)$ with W-CENTER as a target. Moreover, this motion is also the motion along the main trajectory in the control Zone $a(511)a(613)a(715)a(817)$ with the square 817 as a location of the future target, W-CENTER, whose arrival is expected by the tree inspection procedure. As usual, this control Zone was registered in the bottom of the search tree and kept idle until now. Thus 3. ... 511-613 should be considered as a highly time-gaining move. The State 9 generated after 3. ... 511-613 4. 514-515 613:515 5. 818:817 is shown in Fig. 13.

After the futile attempts to continue interception of W-CARRIER by W-CENTER or attack by B-CARRIER, the grammar returns to the State 9. At this moment the tree inspection procedure activates new attack Zone of B-SCOUT from 515 to 817. Among the bundle of such Zones (Fig. 13) the Zone with the most traversable main trajectory $a(515)a(617)a(715)a(817)$ is picked up. After 5. ... 515-617 6. 717-718 617-715 7. 718:715 616:715, the state is exactly the same as State 3 (Fig. 7) generated earlier in the search tree. The only difference is that in the current state there is no W-STATION at 513. As we know the minimax value for the State 3

propagated from the bottom of the search subtree was a draw (0). So, it seems that Black which is currently looking for this value have found one. This, probably, means that after 3. 513-514, Black eventually have found the right variation leading to a draw. But, because of the different location of W-STATION mentioned above we can not just consider this state as the state visited before, terminate this branch, and assign the value. Analogously to the State 3 (on descent), the state evaluation procedure can not assign a definite value to this state, so the branch continues. All the following moves, the CARRIERS race, are exactly the same as in the earlier branch generated from the State 3. The race is complete when both CARRIERS have reached their respective strategic areas. The corresponding State 10 is shown in Fig. 14. The only difference of this state with the State 4 (Fig. 8) is the absence of W-STATION at 513. But this tiny change makes big difference. The motion of W-AS-FIGHTER along the time-gaining trajectory $a(818)a(816)a(311)$ is a simultaneous immediate attack of both B-CENTER and B-AS-FIGHTER. This means that at least one of the targets will be destroyed. The continuation is as follows: 12. 818-816 715-615 (or 12. ... 715-714) 13. 816:311. In both variations W-AS-FIGHTER is destroyed and they are terminated with the value (+1) in favor of White. Thus, despite of this long 25-move(!) resistance, Black achieved nothing. The current minimax value is still in favor of White.

The following climb and branching when Black tries, e.g., most efficiently activate the retreat Zone of B-CENTER from 715 at the upper levels of the search tree or explore different B-SCOUT attack trajectories from 511, does not change the minimax value. The following tree generation does not even yield a "better" (longer) resistance variation than the best variation generated so far. Basically, this longest variation is the optimal variation which is likely to be followed by both sides in the actual combat. In order to generate this branch the grammar used the information, the key networks (W-STATION retreat Zone) learned at the bottom of the search tree in the previously generated non-optimal branches.

The search tree generated by the grammar consists of 152 moves. Obviously, this is a dramatic reduction in comparison with billion-move trees generated by conventional search procedures and still insufficient for solving this problem.

5. DISCUSSION

The example considered in this paper demonstrates the power of the Linguistic Geometry tools that allowed to transfer heuristics discovered in the 2D problem domain of positional games, to another domain of simplified aerospace robotic vehicles. The conventional approaches employing search algorithms with alpha-beta pruning require approximately 30^{25} move search tree to solve this problem, while the tree presented in this paper consists of 152 moves. Moreover, the branching factor of this search, i.e., the average number of moves in each node, is about 1.12(!) while the depth of the search required to solve this problem must be at least 25 moves. This means that the algorithm is actually goal-oriented, i.e., it approaches the goal almost without branching to different directions. Looking at the complexity of the hierarchy of languages which represents each state in the search process, we can suppose that the growth from the problems with the lesser number of agents with limited moving capabilities and smaller 2D operational district (Stilman, 1994b, 1994c) to the current essentially more complex problem is linear with the factor close to one. This means that the complexity of the entire algorithm may be about linear with respect to the length of the input.

At the same time the simplified aerospace navigation problem considered here is still very close to the original chess domain. It is possible to predict that the power of Linguistic Geometry goes far beyond these limits. The definition of the Complex System (Section 3.1) is generic enough to cover a variety of different problem domains. The core component of this definition is the triple X, P, and Rp. Thus, looking at the new problem domain we have to define X, the finite set of points – locations of elements. We do not impose any constraints on this set while the operational district X considered in this paper as well as the original chess board have different extra features, e.g., 2D or 3D connectivity, which is totally unimportant for these problems. Thus, for example, we can consider X as a set of orbits where the elements are in permanent motion with respect to each other. The moving capabilities of elements P in our example, i.e., the binary relations Rp, are non-sophisticated. This is exactly the place for introduction of the variable speed, the gravity impact, the engine impulse duration, etc.

Also, it should be noted that in example considered in this paper we introduced some

additional constraints for the Complex System. These are requirements of the motion alternation for the opposing sides and participation of the only element in each motion. This introduction was done only for a transparent display of ideas and advantages of Linguistic Geometry. The generic definition of the Complex System (Section 3.1) does not include these constraints. The examples where the constraints of single element motion have been relaxed are considered in (Stilman, 1995).

References

- Albus, J. (1991). Outline for a Theory of Intelligence. *IEEE Trans. on Systems, Man and Cybernetics* (pp. 473-509), 3.
- Boddy, M. and Dean, T. (1989). Solving Time-Dependent Planning Problems, *Proc. of the 11th Int. Joint Conf. on AI*, (pp. 979-984).
- Botvinnik, M.M. (1984). *Computers in Chess: Solving Inexact Search Problems*. Springer Series in Symbolic Computation, New York: Springer-Verlag.
- Botvinnik, M., Petriyev, E., Reznitskiy, A., et al. (1983). Application of New Method for Solving Search Problems For Power Equipment Maintenance Scheduling. *Economics and Mathematical Methods* (pp. 1030-1041), 6, (in Russian).
- Chapman, D. (1987). Planning for conjunctive goals. *Artificial Intelligence* 32(3).
- Chomsky, N. (1963). Formal Properties of Grammars. in *Handbook of Mathematical Psychology*, eds. R.Luce, R.Bush, E. Galanter., vol. 2 (pp. 323-418). New York: John Wiley & Sons.
- Chung, J., Liu, J. and Lin, K., (1990). Scheduling Periodic Jobs That Allow Imprecise Results. *IEEE Transactions on Computers*, (pp. 1156-1174), 39(9).
- Drabble, B., (1991) Spacecraft Command and Control Using Artificial Intelligence Techniques, *J. of the British Interplanetary Society*, Vol. 44, (251-254).
- Durfee, E.H. and Lesser, V.R. (1987). Incremental Planning to Control a Blackboard-Based Problem Solver. *Proceedings of the Workshop on Space Telerobotics*, Vol. 3, NASA Jet Propulsion Lab, July, (pp. 91-99).
- Feder, J. (1971). Plex languages. *Information Sciences*, 3: 225-241.
- Fikes, R.E. and Nilsson, N.J. (1971). STRIPS: A New Approach to the Application of Theorem Proving in Problem Solving. *Artificial Intelligence* 2: 189-208.
- Fu, K.S. (1982). *Syntactic Pattern Recognition and Applications*, Prentice Hall, Englewood Cliffs.
- Garcia-Ortiz, A. et al. (1993). Application of Semantic Control to a Class of Pursue-Evader Problems, *Computers and Mathematics with Applications*, 26(5), (pp. 97-124).
- Garey, M.R. and D.S.Johnson D.S. (1991). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., San Francisco.

- Giarratano, J.C., (1991). CLIPS User's Guide, NASA Johnson Space Center, Information Systems Directorate (JSC-25013).
- Ginsburg, S. (1966). *The Mathematical Theory of Context-Free Languages*, McGraw Hill, New York.
- King, R. D. (1993). Rule Based Approach to Hierarchical Grammars for Geometrical Reasoning, Master Thesis, Dept. of Computer Sci., Univ. of Colorado at Denver.
- Knoblock, C.A. (1990). Learning Abstraction Hierarchies for Problem Solving, *Proc. of the 8th AAAI Conf.*, (pp.923-928), Menlo Park, CA.
- Knuth, D.E. (1968). Semantics of Context-Free Languages. *Mathematical Systems Theory*, (pp. 127-146), 2.
- Korf, R.E. (1990). Real-Time Heuristic Search, *Artificial Intelligence*, (pp. 189-211), 42(2-3).
- Leitmann, G., (1990). *Optimization Techniques with Applications to Aerospace Systems*, Academic Press.
- Lesser, V.R., Pavlin, J., and Durfee, E. (1988). Approximate Processing in Real-Time Problem Solving, *AI Magazine*, (pp. 49-62), 9(1).
- Lirov Y., Rodin, E.Y., McElhaney, B.G., and Wilbur, L.W. (1988). Artificial Intelligence Modeling of Control Systems, *Simulation*, (pp. 12-24), 50(1).
- Mathews, E. R. (1993). An Implementation of the Grammars of Trajectories and Zones in C Language, Master Thesis, Dept. of Mathematics, Univ. of Colorado at Denver.
- McAllester, D. and Rosenblitt, D. (1991). Systematic Non-Linear Planning, *Proc. of AAAI-91*, (pp. 634-639).
- McCarthy, J. (1980). Circumscription-A Form of Non-Monotonic Reasoning. *Artificial Intelligence*, (27-39), 13.
- McCarthy, J. and Hayes, P.J. (1969). Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intelligence* (pp. 463-502), 4.
- Mesarovich, M.D. and Takahara Y. (1989). *Abstract Systems Theory*, Berlin: Springer-Verlag.
- Narasimhan, R.N. (1966). Syntax-Directed Interpretation of Classes of Pictures. *Comm. of ACM* (pp. 166-173), 9.
- Nilsson, N.J. (1980). *Principles of Artificial Intelligence*, Palo Alto, CA: Tioga Publ.
- Pavlidis, T. (1977). *Structural Pattern Recognition*, New York: Springer-Verlag.
- Pigeon, A., Howard, G., and Seaton B. (1992) Operational Aspects of Space craft Autonomy, *J. of the British Interplanetary Society*, Vol. 45, (pp. 87-92).
- Reznitskiy, A.I. and Stilman, B. (1983). Use of Method PIONEER in Automating the Planning of Maintenance of Power-Generating Equipment. *Automatics and Remote Control* (pp. 147-153), 11, (in Russian).
- Rodin E. (1988). Semantic Control Theory, *Applied Mathematical Letters*, (pp. 73-78), 1(1).
- Rosenfeld, A. (1979). *Picture Languages, Formal Models for Picture Recognition*, Academic Press.
- Rozenkrantz, D.J. (1969). Programmed Grammars and Classes of Formal Languages, *J. of the ACM* (pp. 107-131), 1.
- Sacerdoti, E.D. (1975). The Nonlinear Nature of Plans, *Proc. Int. Joint Conference on Artificial Intelligence*.
- Shaw, A.C. (1969). A Formal Picture Description Scheme as a Basis for Picture Processing System, *Information and Control* (pp.9-52), 19.
- Shinar, J., (1990). Analysis of Dynamic Conflicts by Techniques of Artificial Intelligence, INRIA Report, Antipolis.
- Simon, H.A. (1980). *The Sciences of the Artificial*, 2-nd ed., The MIT Press, Cambridge, MA.
- Stefik, M. (1981). Planning and meta-planning (MOLGEN: Part 2)," *Artificial Intelligence* (pp. 141-169), 2.
- Stilman, B. (1977). The Computer Learns. in Levy, D., *1976 US Computer Chess Championship* (pp. 83-90). Computer Science Press, Woodland Hills, CA.
- Stilman, B. (1985). Hierarchy of Formal Grammars for Solving Search Problems. In *Artificial Intelligence. Results and Prospects, Proceedings of the International Workshop* (pp. 63-72), Moscow, (in Russian).
- Stilman, B. (1993a). A Linguistic Approach to Geometric Reasoning," *Int. J. Computers and Mathematics with Applications* (pp. 29-57), 26(7).
- Stilman, B. (1993b). Network Languages for Complex Systems, *Int. J. Computers and Mathematics with Applications* (pp. 51-79), 26(8).
- Stilman, B. (1993c). Syntactic Hierarchy for Robotic Systems, *Integrated Computer-Aided Engineering* (pp. 57-81), 1(1).
- Stilman, B. (1993d). A Formal Language for Hierarchical Systems Control, *Languages of Design* (333-356), 1(4).
- Stilman, B. (1994a). Translations of Network Languages. *Int. J. Computers and Mathematics with Applications* (pp. 65-98), 27(2).
- Stilman, B. (1994b). A Formal Model for Heuristic Search. *Proc. of the 22nd Annual ACM Computer Science Conf.*, (pp. 380-389), March 8-10, Phoenix, AZ.
- Stilman, B., (1994c). A Linguistic Geometry for Space Applications, *Proc. of the 1994 Goddard Conference on Space Applications of Artificial Intelligence*, (pp. 87-101), NASA Goddard Space Flight Center, Greenbelt, MD, USA, May 1994.
- Stilman, B., (1995). Multiagent Air Combat with Synchronous Motions, Symposium on Linguistic Geometry and Semantic Control, *Proc. of the First World Congress on Intelligent Manufacturing: Processes and Systems*, Mayaguez, Puerto Rico, Feb. 1995, (to appear).
- Strosnider, J.K., Paul, C.J. (1994). A Structured View of Real-Time Problem Solving, *AI Magazine*, (45-66), 15(2).

*OMIT TO
END*

Goddard Space Flight Center

Conference Bibliography 1986-1994



National Aeronautics and
Space Administration
**Scientific and Technical
Information Office**

USING THE BIBLIOGRAPHY

The first portion of this bibliography contains citations (with abstracts, when available) to unclassified literature contained in the NASA STI Database. These citations also appeared in issues of the abstract journals *Scientific and Technical Aerospace Reports (STAR)* or from open literature. The citations appear in ascending accession number order.

A second section provides a subject index to the citations.

AVAILABILITY OF CITED PUBLICATIONS

NASA-sponsored report literature is available to NASA personnel, NASA contractors, and other United States Government agencies and their contractors, through the STI Office, facilitated by the NASA Center for AeroSpace Information. For further information on the availability and cost of any item, contact:

NASA Access Help Desk
NASA Center for AeroSpace Information
800 Elkridge Landing Road
Linthicum, MD 21090-2934
Telephone: (301) 621-0390
Fax: (301) 621-0134
Internet: help@sti.nasa.gov

This publication was prepared by the NASA Center for AeroSpace Information,
800 Elkridge Landing Road, Linthicum Heights, MD 21090-2934, (301) 621-0390.

TYPICAL REPORT CITATION AND ABSTRACT

**NASA SPONSORED
ON MICROFICHE**

ACCESSION NUMBER → N90-10479*# Environmental Research Inst. of Michigan, Ann Arbor. Advanced Concepts Div. ← CORPORATE SOURCE

TITLE → SPATIAL CHARACTERIZATION OF ACID RAIN STRESS IN CANADIAN SHIELD LAKES Final Report

AUTHORS → F. J. TANIS and E. M. MARSHALL Mar. 1989 176 p ← PUBLICATION DATE

CONTRACT NUMBER → (Contract NAS5-28779)

REPORT NUMBERS → (NAS-CR-183446; NAS 1.26:183446; E-189400-39-F) Avail:

AVAILABILITY SOURCE → NTIS HC A09/MF A01 CSCL 13B ← COSATI CODE

PRICE CODE → The lake acidification in Northern Ontario was investigated using LANDSAT TM to sense lake volume reflectance and also to provide important vegetation and terrain characteristics. The purpose of this project was to determine the ability of LANDSAT to assess water quality characteristics associated with lake acidification. Results demonstrate that a remote sensor can discriminate lake clarity based upon reflection. The basic hypothesis is that seasonal and multi-year changes in lake optical transparency are indicative of sensitivity to acidic deposition. In many acid-sensitive lakes optical transparency is controlled by the amount of dissolved organic carbon present. Seasonal changes in the optical transparency of lakes can potentially provide an indication of the stress due to acid deposition and loading.
Author

TYPICAL JOURNAL ARTICLE CITATION AND ABSTRACT

NASA SPONSORED

ACCESSION NUMBER → A90-21682* Reading Univ. (England). ← CORPORATE SOURCE

TITLE → COMPARISON OF DATA FROM THE SCANNING MULTIFREQUENCY MICROWAVE RADIOMETER (SMMR) WITH DATA FROM THE ADVANCED VERY HIGH RESOLUTION RADIOMETER (AVHRR) FOR TERRESTRIAL ENVIRONMENTAL MONITORING - AN OVERVIEW

AUTHORS → J. R. G. TOWNSHEND (Reading, University, England), B. J. CHOUDHURY, C. J. TUCKER (NASA, Goddard Space Flight Center, Greenbelt, MD), L. GIDDINGS (Instituto Nacional de Investigaciones sobre Recursos Bióticos, Veracruz, Mexico), C. O. JUSTICE (Maryland, University, College Park) et al. International Journal of Remote Sensing (ISSN 0143-1161), vol. 10, Oct. 1989, p. 1687-1690. ← AUTHORS' AFFILIATION ← JOURNAL TITLE

refs
Copyright
Comparison between the microwave polarized difference temperature (MPDT) derived from 37 GHz band data and the normalized difference vegetation index (NDVI) derived from near-infrared and red bands, from several empirical investigations are summarized. These indicate the complementary character of the two measures in environmental monitoring. Overall the NDVI is more sensitive to green leaf activity, whereas the MPDT appears also to be related to other elements of the above-ground biomass. Monitoring of hydrological phenomena is carried out much more effectively by the MPDT. Further work is needed to explain spectral and temporal variation in MPDT both through modelling and field experiments.
Author

SELECTED BIBLIOGRAPHY

N88-30330*# National Aeronautics and Space Administration.
Goddard Space Flight Center, Greenbelt, MD.

THE 1988 GODDARD CONFERENCE ON SPACE APPLICATIONS OF ARTIFICIAL INTELLIGENCE

JAMES RASH, ed. and PETER HUGHES, ed. Aug. 1988
437 p Conference held in Greenbelt, Md., 24 May 1988
Sponsored by NASA, Washington, D.C.
(NASA-CP-3009; REPT-88B0212; NAS 1.55:3009) Avail:
CASI HC A19/MF A04

This publication comprises the papers presented at the 1988 Goddard Conference on Space Applications of Artificial Intelligence held at the NASA/Goddard Space Flight Center, Greenbelt, Maryland on May 24, 1988. The purpose of this annual conference is to provide a forum in which current research and development directed at space applications of artificial intelligence can be presented and discussed. The papers in these proceedings fall into the following areas: mission operations support, planning and scheduling; fault isolation/diagnosis; image processing and machine vision; data management; modeling and simulation; and development tools/methodologies.

N88-30331*# National Aeronautics and Space Administration.
Lyndon B. Johnson Space Center, Houston, TX.

AN INTELLIGENT TRAINING SYSTEM FOR SPACE SHUTTLE FLIGHT CONTROLLERS

R. BOWEN LOFTIN (Houston Univ., TX.), LUI WANG (National Aeronautics and Space Administration. Lyndon B. Johnson Space Center, Houston, TX.), PAUL BAFFES (National Aeronautics and Space Administration. Lyndon B. Johnson Space Center, Houston, TX.), and GRACE HUA (Computer Sciences Corp., Houston, Tex.) In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 3-15 Aug. 1988
Avail: CASI HC A03/MF A04

An autonomous intelligent training system which integrates expert system technology with training/teaching methodologies is described. The system was designed to train Mission Control Center (MCC) Flight Dynamics Officers (FDOs) to deploy a certain type of satellite from the Space Shuttle. The Payload-assist module Deploys/Intelligent Computer-Aided Training (PD/ICAT) system consists of five components: a user interface, a domain expert, a training session manager, a trainee model, and a training scenario generator. The interface provides the trainee with information of the characteristics of the current training session and with on-line help. The domain expert (DeplEx for Deploy Expert) contains the rules and procedural knowledge needed by the FDO to carry out the satellite deploy. The DeplEx also contains mal-rules which permit the identification and diagnosis of common errors made by the trainee. The training session manager (TSM) examines the actions of the trainee and compares them with the actions of DeplEx in order to determine appropriate responses. A trainee model is developed for each individual using the system. The model includes a history of the trainee's interactions with the training system and provides evaluative data on the trainee's current skill level. A training scenario generator (TSG) designs appropriate training exercises for each trainee based on the trainee model and the training goals. All of the expert system components of PD/ICAT communicate via a common

blackboard. The PD/ICAT is currently being tested. Ultimately, this project will serve as a vehicle for developing a general architecture for intelligent training systems together with a software environment for creating such systems. Author

N88-30332*# National Aeronautics and Space Administration.
Goddard Space Flight Center, Greenbelt, MD.

ARTIFICIAL INTELLIGENCE COSTS, BENEFITS, RISKS FOR SELECTED SPACECRAFT GROUND SYSTEM AUTOMATION SCENARIOS

WALTER F. TRUSzkowski (National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.), BARRY G. SILVERMAN (George Washington Univ., Washington, DC.), MARTHA KAHN (Intellitek, Inc., Rockville, MD.), and HENRY HEXMOOR (Intellitek, Inc., Rockville, Md.) In its The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 17-31 Aug. 1988
Avail: CASI HC A03/MF A04

In response to a number of high-level strategy studies in the early 1980s, expert systems and artificial intelligence (AI/ES) efforts for spacecraft ground systems have proliferated in the past several years primarily as individual small to medium scale applications. It is useful to stop and assess the impact of this technology in view of lessons learned to date, and hopefully, to determine if the overall strategies of some of the earlier studies both are being followed and still seem relevant. To achieve that end four idealized ground system automation scenarios and their attendant AI architecture are postulated and benefits, risks, and lessons learned are examined and compared. These architectures encompass: (1) no AI (baseline), (2) standalone expert systems, (3) standardized, reusable knowledge base management systems (KBMS), and (4) a futuristic unattended automation scenario. The resulting artificial intelligence lessons learned, benefits, and risks for spacecraft ground system automation scenarios are described. Author

N88-30333*# National Aeronautics and Space Administration.
Ames Research Center, Moffett Field, CA.

A SHARED-WORLD CONCEPTUAL MODEL FOR INTEGRATING SPACE STATION LIFE SCIENCES TELESCIENCE OPERATIONS

VICKI JOHNSON (Research Inst. for Advanced Computer Science, Moffett Field, CA.) and JOHN BOSLEY (Bionetics Corp., Moffett Field, Calif.) In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 33-44 Aug. 1988
Avail: CASI HC A03/MF A04

Mental models of the Space Station and its ancillary facilities will be employed by users of the Space Station as they draw upon past experiences, perform tasks, and collectively plan for future activities. The operational environment of the Space Station will incorporate telescience, a new set of operational modes. To investigate properties of the operational environment, distributed users, and the mental models they employ to manipulate resources while conducting telescience, an integrating shared-world conceptual model of Space Station telescience is proposed. The model comprises distributed users and resources (active elements); agents who mediate interactions among these elements on the basis of intelligent processing of shared information; and telescience protocols which structure the

interactions of agents as they engage in cooperative, responsive interactions on behalf of users and resources distributed in space and time. Examples from the life sciences are used to instantiate and refine the model's principles. Implications for transaction management and autonomy are discussed. Experiments employing the model are described which the authors intend to conduct using the Space Station Life Sciences Telescience Testbed currently under development at Ames Research Center.

Author

N88-30334*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

ARTIFICIAL INTELLIGENCE IN A MISSION OPERATIONS AND SATELLITE TEST ENVIRONMENT
CARL BUSSE In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 45-57 Aug. 1988

Avail: CASI HC A03/MF A04

A Generic Mission Operations System using Expert System technology to demonstrate the potential of Artificial Intelligence (AI) automated monitor and control functions in a Mission Operations and Satellite Test environment will be developed at the National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory (JPL). Expert system techniques in a real time operation environment are being studied and applied to science and engineering data processing. Advanced decommutation schemes and intelligent display technology will be examined to develop imaginative improvements in rapid interpretation and distribution of information. The Generic Payload Operations Control Center (GPOCC) will demonstrate improved data handling accuracy, flexibility, and responsiveness in a complex mission environment. The ultimate goal is to automate repetitious mission operations, instrument, and satellite test functions by the applications of expert system technology and artificial intelligence resources and to enhance the level of man-machine sophistication.

Author

N88-30335*# National Aeronautics and Space Administration. Lyndon B. Johnson Space Center, Houston, TX.

AUTOMATED SPACE VEHICLE CONTROL FOR RENDEZVOUS PROXIMITY OPERATIONS

ROBERT N. LEA In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 59-66 Aug. 1988

Avail: CASI HC A02/MF A04

Rendezvous during the unmanned space exploration missions, such as a Mars Rover/Sample Return will require a completely automatic system from liftoff to docking. A conceptual design of an automated rendezvous, proximity operations, and docking system is being implemented and validated at the Johnson Space Center (JSC). The emphasis is on the progress of the development and testing of a prototype system for control of the rendezvous vehicle during proximity operations that is currently being developed at JSC. Fuzzy sets are used to model the human capability of common sense reasoning in decision making tasks and such models are integrated with the expert systems and engineering control system technology to create a system that performs comparably to a manned system.

Author

N88-30336*# Ford Aerospace and Communications Corp., Seabrook, MD.

AUTOMATED SATELLITE CONTROL IN ADA

ALLAN JAWORSKI and J. T. THOMPSON In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 67-75 Aug. 1988 Avail: CASI HC A02/MF A04

The Advanced Ground Segment, a prototype satellite/payload operations control center workstation, which represents an evolutionary effort to improve the automation of control centers while improving software practices and supporting distributed control center functions, is described. Multiple levels of automation are supported through a rule-based control strategy. The architecture provides the necessary interfaces and modularity for future inclusion of more sophisticated control strategies.

Author

N88-30337*# Martin Marietta Corp., Denver, CO. Information and Communication Systems.

CONTINGENCY RESCHEDULING OF SPACECRAFT OPERATIONS

DANIEL L. BRITT, AMY L. GEOFFROY, and JOHN R. GOHRING In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 79-90 Aug. 1988

Avail: CASI HC A03/MF A04

Spacecraft activity scheduling was a focus of attention in artificial intelligence recently. Several scheduling systems were devised which more-or-less successfully address various aspects of the activity scheduling problem, though most of these are not yet mature, with the notable exception of NASA's ESP. Few current scheduling systems, however, make any attempt to deal fully with the problem of modifying a schedule in near-real-time in the event of contingencies which may arise during schedule execution. These contingencies can include resources becoming unavailable unpredictably, a change in spacecraft conditions or environment, or the need to perform an activity not scheduled. In these cases it becomes necessary to repair an existing schedule, disrupting ongoing operations as little as possible. Normal scheduling is just a part of that which must be accomplished during contingency rescheduling. A prototype system named MAESTRO was developed for spacecraft activity scheduling. MAESTRO is briefly described with a focus on recent work in the area of real-time contingency handling. Included is a discussion of some of the complexities of the scheduling problem and how they affect contingency rescheduling, such as temporal constraints between activities, activities which may be interrupted and continued in any of several ways, and different ways to choose a resource complement which will allow continuation of an activity. Various heuristics used in MAESTRO for contingency rescheduling is discussed, as are operational concerns such as interaction of the scheduler with spacecraft subsystems controllers.

Author

N88-30338*# Computer Sciences Corp., Baltimore, MD. Astronomy Programs.

KNOWLEDGE BASED TOOLS FOR HUBBLE SPACE TELESCOPE PLANNING AND SCHEDULING: CONSTRAINTS AND STRATEGIES

GLENN MILLER (Computer Sciences Corp., Baltimore, MD.), MARK JOHNSTON (Space Telescope Science Inst., Baltimore, MD.), SHON VICK (Space Telescope Science Inst., Baltimore, MD.), JEFF SPONSLER (Space Telescope Science Inst., Baltimore, Md.), and KELLY LINDEMAYER In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 91-106 Aug. 1988

Avail: CASI HC A03/MF A04

The Hubble Space Telescope (HST) presents an especially challenging scheduling problem since a year's observing program encompasses tens of thousands of exposures facing numerous coupled constraints. Recent progress in the development of planning and scheduling tools is discussed which augment the existing HST ground system. General methods for representing activities, constraints, and constraint satisfaction, and time segmentation were implemented in a scheduling testbed. The testbed permits planners to evaluate optimal scheduling time intervals, calculate resource usage, and to generate long and medium range plans. Graphical displays of activities, constraints, and plans are an important feature of the system. High-level scheduling strategies using rule based and neural net approaches were implemented. Author

N88-30339*# Computer Sciences Corp., Baltimore, MD. Astronomy Programs.

THE PROPOSAL ENTRY PROCESSOR: TELESCIENCE APPLICATIONS FOR HUBBLE SPACE TELESCOPE SCIENCE OPERATIONS

ROBERT JACKSON (Computer Sciences Corp., Baltimore, MD.), MARK JOHNSTON (Space Telescope Science Inst., Baltimore, MD.), GLENN MILLER (Space Telescope Science Inst., Baltimore, MD.), KELLY LINDENMAYER (Space Telescope Science Inst., Baltimore, MD.), PATRICIA MONGER (Space Telescope Science Inst., Baltimore, MD.), SHON VICK (Space Telescope Science Inst., Baltimore, MD.), ROBIN LERNER (Space Telescope Science Inst., Baltimore, MD.), and JOEL RICHON (Space Telescope Science Inst., Baltimore, Md.) In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 107-123 Aug. 1988

Avail: CASI HC A03/MF A04

The Proposal Entry Processor (PEP) System supports the submission, entry, technical evaluation review, selection and implementation of Hubble Space Telescope (HST) observing proposals. The PEP system is described concentrating on features which illustrate principles of telescience as applied to the HST. These principles are applicable to other observatories, both space and ground based. The PEP proposal forms allow a scientist to specify scientific objectives without becoming needlessly involved in implementation details. The Remote Proposal Submission System (RPSS) allows proposers to submit proposals electronically via Telenet, SPAN, and other networks. The RPSS performs syntax and semantic checks on proposals. The PEP uses a fourth generation database system to store proposal information and to allow general queries and reports. The Transformation subsystem uses an expert system written in OPS5 to cast a scientific description of an observing program into parameters used by the planning and scheduling system. The TACOS system is a natural language database which supports the proposal selection process. Technical evaluations for resource usage and duplicate science are performed using rulebased systems.

Author

N88-30340*# Computer Sciences Corp., Beltsville, MD.

CANDIDATE FUNCTIONS FOR ADVANCED TECHNOLOGY IMPLEMENTATION IN THE COLUMBUS MISSION PLANNING ENVIRONMENT

AUDREY LOOMIS (Computer Sciences Corp., Beltsville, MD.) and ALBRECHT KELLNER (Messerschmitt-Boelkow-Blohm G.m.b.H., Bremen, West Germany) In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space

Applications of Artificial Intelligence p 125-139 Aug. 1988

Prepared in cooperation with Entwicklungsring Nord, Bremen, Fed. Republic of Germany

Avail: CASI HC A03/MF A04

The Columbus Project is the European Space Agency's contribution to the International Space Station program. Columbus is planned to consist of three elements (a laboratory module attached to the Space Station base, a man-tended freeflyer orbiting with the Space Station base, and a platform in polar orbit). System definition and requirements analysis for Columbus are underway, scheduled for completion in mid-1990. An overview of the Columbus mission planning environment and operations concept as currently defined is given, and some of the challenges presented to software maintainers and ground segment personnel during mission operators are identified. The use of advanced technologies in system implementation is being explored. Both advantages of such solutions and potential problems they present are discussed, and the next steps to be taken by Columbus before targeting any functions for advanced technology implementation are summarized. Several functions in the mission planning process were identified as candidates for advanced technology implementation. These range from expert interaction with Columbus' data bases through activity scheduling and near-real-time response to departures from the planned timeline. Each function is described, and its potential for advanced technology implementation briefly assessed. Author

N88-30341**# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

A RULE-BASED SYSTEMS APPROACH TO SPACE-CRAFT COMMUNICATIONS CONFIGURATION OPTIMIZATION

JAMES L. RASH, YEN F. WONG, and JAMES J. CIEPLAK In its The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 141-153 Aug. 1988

Avail: CASI HC A03/MF A04

An experimental rule-based system for optimizing user spacecraft communications configurations was developed at NASA to support mission planning for spacecraft that obtain telecommunications services through NASA's Tracking and Data Relay Satellite System. Designated Expert for Communications Configuration Optimization (ECCO), and implemented in the OPS5 production system language, the system has shown the validity of a rule-based systems approach to this optimization problem. The development of ECCO and the incremental optimization method on which it is based are discussed. A test case using hypothetical mission data is included to demonstrate the optimization concept. Author

N88-30342*# Ford Aerospace and Communications Corp., College Park, MD.

INTEGRATED RESOURCE SCHEDULING IN A DISTRIBUTED SCHEDULING ENVIRONMENT

DAVID ZOCH and GARDINER HALL In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 155-172 Aug. 1988

Avail: CASI HC A03/MF A04

The Space Station era presents a highly-complex multi-mission planning and scheduling environment exercised over a highly distributed system. In order to automate the scheduling process, customers require a mechanism for communicating their scheduling requirements to NASA. A request language that a remotely-located customer can use to specify his scheduling requirements to a NASA scheduler, thus automating the

customer-scheduler interface, is described. This notation, Flexible Envelope-Request Notation (FERN), allows the user to completely specify his scheduling requirements such as resource usage, temporal constraints, and scheduling preferences and options. The FERN also contains mechanisms for representing schedule and resource availability information, which are used in the inter-scheduler inconsistency resolution process. Additionally, a scheduler is described that can accept these requests, process them, generate schedules, and return schedule and resource availability information to the requester. The Request-Oriented Scheduling Engine (ROSE) was designed to function either as an independent scheduler or as a scheduling element in a network of schedulers. When used in a network of schedulers, each ROSE communicates schedule and resource usage information to other schedulers via the FERN notation, enabling inconsistencies to be resolved between schedulers. Individual ROSE schedules are created by viewing the problem as a constraint satisfaction problem with a heuristically guided search strategy.

Author

N88-30343*# Westinghouse Electric Corp., Seabrook, MD.

**MOORE: A PROTOTYPE EXPERT SYSTEM FOR
DIAGNOSING SPACECRAFT PROBLEMS**

KATHERINE HOWLIN, JERRY WEISSERT, and KERRY KRANTZ In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 175-190 Aug. 1988

Avail: CASI HC A03/MF A04

MOORE is a rule-based, prototype expert system that assists in diagnosing operational Tracking and Data Relay Satellite (TDRS) problems. It is intended to assist spacecraft engineers at the TDRS ground terminal in trouble shooting problems that are not readily solved with routine procedures, and without expert counsel. An additional goal of the prototype system is to develop in-house expert system and knowledge engineering skills. The prototype system diagnoses antenna pointing and earth pointing problems that may occur within the TDRS Attitude Control System (ACS). Plans include expansion to fault isolation of problems in the most critical subsystems of the TDRS spacecraft. Long term benefits are anticipated with use of an expert system during future TDRS programs with increased mission support time, reduced problem solving time, and retained expert knowledge and experience. Phase 2 of the project is intended to provide NASA the necessary expertise and capability to define requirements, evaluate proposals, and monitor the development progress of a highly competent expert system for NASA's Tracking Data Relay Satellite. Phase 2 also envisions addressing two unexplored applications for expert systems, spacecraft integration and tests (I and T) and support to launch activities. The concept, goals, domain, tools, knowledge acquisition, developmental approach, and design of the expert system. It will explain how NASA obtained the knowledge and capability to develop the system in-house without assistance from outside consultants. Future plans will also be presented.

Author

N88-30344*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

ACHIEVING REAL-TIME PERFORMANCE IN FIESTA

WILLIAM WILKINSON (Stanford Telecommunications, Inc., Reston, VA.), NADINE HAPPELL (Stanford Telecommunications, Inc., Reston, VA.), STEVE MIKSELL (Stanford Telecommunications, Inc., Reston, VA.), ROBERT QUILLIN (Stanford Telecommunications, Inc., Reston, Va.), and CANDACE CARLISLE In its The 1988 Goddard Conference on Space

Applications of Artificial Intelligence p 191-205 Aug. 1988
(Contract NAS5-31500)

Avail: CASI HC A03/MF A04

The Fault Isolation Expert System for TDRSS Applications (FIESTA) is targeted for operation in a real-time online environment. Initial stages of the prototype development concentrated on acquisition and representation of the knowledge necessary to isolate faults in the TDRSS Network. Recent efforts focused on achieving real-time performance including: a discussion of the meaning of FIESTA real-time requirements, determination of performance levels (benchmarking) and techniques for optimization. Optimization techniques presented include redesign of critical relations, filtering of redundant data and optimization of patterns used in rules. Results are summarized.

Author

N88-30345*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

MISSION TELEMETRY SYSTEM MONITOR: A REAL-TIME KNOWLEDGE-BASED SYSTEM

SAMIH A. MOUNEIMNE In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 207-212 Aug. 1988

Avail: CASI HC A02/MF A04

The Galileo Mission Telemetry System (MTS) has a cluster of computer subsystems configured as a star network. The MTS handles the real-time processing of spacecraft telemetry and ground monitor data. Large volumes of status and fault messages are generated as a result of changes in the system environment. These messages are triggered by the conditions that exist on any one particular subsystem or device. The order of message generation is in time sequence and does not always correlate to the function sequence of active processes. A significant number of messages provide context with varying degrees of uncertainty. As such, highly skilled telemetry controllers are required to regularly go through high volumes of messages generated by the MTS to identify, diagnose, and isolate faults. A knowledge-based system prototype is being developed to monitor the Galileo Mission Telemetry System performance. The system design approach features temporal reasoning, uncertainty management, and intelligent graphic user interfaces.

Author

N88-30346*# National Bureau of Standards, Gaithersburg, MD.
**LOW LEVEL IMAGE PROCESSING TECHNIQUES
USING THE PIPELINE IMAGE PROCESSING ENGINE
IN THE FLIGHT TELEROBOTIC SERVICER**

MARILYN NASHMAN and KAREN J. CHACONAS In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 215-229 Aug. 1988

Avail: CASI HC A03/MF A04

The sensory processing system for the NASA/NBS Standard Reference Model (NASREM) for telerobotic control is described. This control system architecture was adopted by NASA of the Flight Telerobotic Servicer. The control system is hierarchically designed and consists of three parallel systems: task decomposition, world modeling, and sensory processing. The Sensory Processing System is examined, and in particular the image processing hardware and software used to extract features at low levels of sensory processing for tasks representative of those envisioned for the Space Station such as assembly and maintenance are described.

Author

N88-30347*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

AUTONOMOUS IMAGE DATA REDUCTION BY ANALYSIS AND INTERPRETATION

SUSAN EBERLEIN, GIGI YATES, and NILES RITTER In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 231-242 Aug. 1988

Avail: CASI HC A03/MF A04

Image data is a critical component of the scientific information acquired by space missions. Compression of image data is required due to the limited bandwidth of the data transmission channel and limited memory space on the acquisition vehicle. This need becomes more pressing when dealing with multispectral data where each pixel may comprise 300 or more bytes. An autonomous, real time, on-board image analysis system for an exploratory vehicle such as a Mars Rover is developed. The completed system will be capable of interpreting image data to produce reduced representations of the image, and of making decisions regarding the importance of data based on current scientific goals. Data from multiple sources, including stereo images, color images, and multispectral data, are fused into single image representations. Analysis techniques emphasize artificial neural networks. Clusters are described by their outlines and class values. These analysis and compression techniques are coupled with decision making capacity for determining importance of each image region. Areas determined to be noise or uninteresting can be discarded in favor of more important areas. Thus limited resources for data storage and transmission are allocated to the most significant images.

Author

N88-30348*# Maryland Univ., College Park, MD. Robotics Lab.

AN AUTOMATED COMPUTERIZED VISION TECHNIQUE FOR DETERMINATION OF THREE-DIMENSIONAL OBJECT GEOMETRY

PEN-TAI CHIANG, JACKSON C. S. YANG, and V. PAVLIN In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 243-257 Aug. 1988

(Contract NSF MSM-86-15187)

Avail: CASI HC A03/MF A04

It is very important to determine three dimensional geometry of objects quickly in various military, space, construction, and industrial applications. An automatic scheme to obtain three dimensional geometry of objects by employing only one camera is presented. At present, this technique is applicable to a limited category of objects, satisfying the following constraints: they are flat-surfaces, and all the vertex points have to be recognized as corner points of the two dimensional image. The scheme consists of corner detection, data communication, camera calibration techniques and point searching and matching, edge cancelation, and creation procedures. An L shaped model is chosen as a test object. Experimental results demonstrated the reconstruction of this object geometry within 5 mm discrepancy. This scheme is quite convenient, efficient to use and can be applied to a wide range of problems in the real world.

Author

N88-30349*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

AN INTERACTIVE TESTBED FOR DEVELOPMENT OF EXPERT TOOLS FOR PATTERN RECOGNITION

STEPHEN W. WHARTON In its The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 259-273 Aug. 1988

Avail: CASI HC A03/MF A04

The initial implementation of an interactive testbed for development of expert system applications in image processing, i.e., a toolbox of procedures designed to facilitate the capture of expert knowledge for region grouping and analysis is described. The user can elect to interactively enter commands (via a command interpreter) for region manipulation to, in effect, simulate the actions of a hypothetical expert system. The user can then incorporate any rules and procedures as derived from interactive experimentation into customized region processing procedures using the library of utility functions. An iterative technique based on image pyramids is used to compute the initial region segmentation without the use of process parameters. These regions can then be interactively examined and manipulated using the command interpreter.

Author

N88-30350*# George Mason Univ., Fairfax, VA. Dept. of Computer Science.

PARALLEL AND DISTRIBUTED COMPUTATION FOR FAULT-TOLERANT OBJECT RECOGNITION

HARRY WECHSLER In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 275-293 Aug. 1988

Avail: CASI HC A03/MF A04

The distributed associative memory (DAM) model is suggested for distributed and fault-tolerant computation as it relates to object recognition tasks. The fault-tolerance is with respect to geometrical distortions (scale and rotation), noisy inputs, occlusion/overlap, and memory faults. An experimental system was developed for fault-tolerant structure recognition which shows the feasibility of such an approach. The approach is further extended to the problem of multisensory data integration and applied successfully to the recognition of colored polyhedral objects.

Author

N88-30351*# Wayne State Univ., Detroit, MI.

RANGE DATA DESCRIPTION BASED ON MULTIPLE CHARACTERISTICS

EZZET AL-HUJAZI (Wayne State Univ., Detroit, MI.) and ARUN SOOD (George Mason Univ., Fairfax, Va.) In NASA, Goddard Space Flight Center, The 1988 Goddard Conf. on Space Applications of Artificial Intelligence p 295-309 Aug. 1988

Avail: CASI HC A03/MF A04

An algorithm for describing range images based on Mean curvature (H) and Gaussian curvature (K) is presented. Range images are unique in that they directly approximate the physical surfaces of a real world 3-D scene. The curvature parameters are derived from the fundamental theorems of differential geometry and provides visible invariant pixel labels that can be used to characterize the scene. The sign of H and K can be used to classify each pixel into one of eight possible surface types. Due to the sensitivity of these parameters to noise the resulting HK-sing map does not directly identify surfaces in the range images and must be further processed. A region growing algorithm based on modeling the scene points with a Markov Random Field (MRF) of variable neighborhood size and edge models is suggested. This approach allows the integration of information from multiple characteristics in an efficient way. The performance of the proposed algorithm on a number of synthetic and real range images is discussed.

Author

N88-30352*# National Aeronautics and Space Administration.
Goddard Space Flight Center, Greenbelt, MD.

THE SECOND GENERATION INTELLIGENT USER INTERFACE FOR THE CRUSTAL DYNAMICS DATA INFORMATION SYSTEM

NICHOLAS SHORT, JR. and SCOTT L. WATTAWA (Science Applications Research, Greenbelt, Md.) In its The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 313-327 Aug. 1988

Avail: CASI HC A03/MF A04

For the past decade, operations and research projects that support a major portion of NASA's overall mission have experienced a dramatic increase in the volume of generated data and resultant information that is unparalleled in the history of the agency. The effect of such an increase is that most of the science and engineering disciplines are undergoing an information glut, which has occurred, not only because of the amount, but also because of the type of data being collected. This information glut is growing exponentially and is expected to grow for the foreseeable future. Consequently, it is becoming physically and intellectually impossible to identify, access, modify, and analyze the most suitable information. Thus, the dilemma arises that the amount and complexity of information has exceeded and will continue to exceed, using present information systems, the ability of all the scientists and engineers to understand and take advantage of this information. As a result of this information problem, NASA has initiated the Intelligent Data Management (IDM) project to design and develop Advanced Information Management Systems (AIMS). The first effort of the Project was the prototyping of an Intelligent User Interface (IUI) to an operational scientific database using expert systems, natural language processing, and graphics technologies. An overview of the IUI formulation and development for the second phase is presented.

Author

N88-30353*# National Aeronautics and Space Administration.
Goddard Space Flight Center, Greenbelt, MD.

**SPACELAB DATA PROCESSING FACILITY (SLDPF)
QUALITY ASSURANCE (QA)/DATA ACCOUNTING (DA)
EXPERT SYSTEMS: TRANSITION FROM PROTOTYPES
TO OPERATIONAL SYSTEMS**

LISA BASILE In its The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 329-341 Aug. 1988
Avail: CASI HC A03/MF A04

The SLDPF is responsible for the capture, quality monitoring processing, accounting, and shipment of Spacelab and/or Attached Shuttle Payloads (ASP) telemetry data to various user facilities. Expert systems will aid in the performance of the quality assurance and data accounting functions of the two SLDPF functional elements: the Spacelab Input Processing System (SIPS) and the Spacelab Output Processing System (SOPS). Prototypes were developed for each as independent efforts. The SIPS Knowledge System Prototype (KSP) used the commercial shell OPS5+ on an IBM PC/AT; the SOPS Expert System Prototype used the expert system shell CLIPS implemented on a Macintosh personal computer. Both prototypes emulate the duties of the respective QA/DA analysts based upon analyst input and predetermined mission criteria parameters, and recommended instructions and decisions governing the reprocessing, release, or holding for further analysis of data. These prototypes demonstrated feasibility and high potential for operational systems. Increase in productivity, decrease of tedium, consistency, concise historical records, and a training tool for new analyses were the principal advantages. An operational

configuration, taking advantage of the SLDPF network capabilities, is under development with the expert systems being installed on SUN workstations. This new configuration in conjunction with the potential of the expert systems will enhance the efficiency, in both time and quality, of the SLDPF's release of Spacelab/AST data products.

Author

N88-30354*# National Aeronautics and Space Administration.
Goddard Space Flight Center, Greenbelt, MD.

**AUTOMATED CATALOGING AND
CHARACTERIZATION OF SPACE-DERIVED DATA**

WILLIAM J. CAMPBELL (National Aeronautics and Space Administration, Goddard Space Flight Center, Greenbelt, MD.), LARRY ROELOFS (Computer Technology Associates, Inc., McLean, VA.), and MICHAEL GOLDBERG (Mitre Corp., McLean, Va.) In its The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 343-354 Aug. 1988
Avail: CASI HC A03/MF A04

One of the most significant technical issues that NASA must address and resolve is the problem of managing the enormous amounts of scientific and engineering data that will be generated by the next generation of remote sensing systems such as the Hubble Space Telescope (HST) and the Earth Observing System (EOS). The amount of data these sensors are expected to produce will be orders of magnitude greater than NASA has ever experienced. Consequently new solutions must be developed for managing, accessing, and automatically inputting the data into a database in some expressive fashion that will provide a meaningful understanding and effective utilization of this data in a multidisciplinary environment. Presently, scientific data provided by satellites and other sources are processed, cataloged, and archived according to narrow mission or project-specific requirements with little regard to the semantics of the overall research. Scientists therefore lack knowledge of or access to potentially valuable data outside their own field. What is needed is an innovative approach that will allow collected data to be automatically cataloged, characterized, and managed in a domain-community. A concept and design approach that employs expert system-based knowledge controllers combined with advanced spatial database systems and graphical data structures is discussed.

Author

N88-30355*# LNK Corp., Riverdale, MD.

**A DESIGN FOR A GROUND-BASED DATA
MANAGEMENT SYSTEM**

BARBARA A. LAMBIRD and DAVID LAVINE In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 355-369 Aug. 1988

Avail: CASI HC A03/MF A04

An initial design for a ground-based data management system which includes intelligent data abstraction and cataloging is described. The large quantity of data on some current and future NASA missions leads to significant problems in providing scientists with quick access to relevant data. Human screening of data for potential relevance to a particular study is time-consuming and costly. Intelligent databases can provide automatic screening when given relevant scientific parameters and constraints. The data management system would provide, at a minimum, information of availability of the range of data, the type available, specific time periods covered together with data quality information, and related sources of data. The system would inform the user about the primary types of screening, analysis, and methods of presentation available to the user. The

system would then aid the user with performing the desired tasks, in such a way that the user need only specify the scientific parameters and objectives, and not worry about specific details for running a particular program. The design contains modules for data abstraction, catalog plan abstraction, a user-friendly interface, and expert systems for data handling, data evaluation, and application analysis. The emphasis is on developing general facilities for data representation, description, analysis, and presentation that will be easily used by scientists directly, thus bypassing the knowledge acquisition bottleneck. Expert system technology is used for many different aspects of the data management system, including the direct user interface, the interface to the data analysis routines, and the analysis of instrument status.

Author

N88-30356*# National Aeronautics and Space Administration, Marshall Space Flight Center, Huntsville, AL.

AUTOMATIC MATHEMATICAL MODELING FOR REAL TIME SIMULATION SYSTEM

CAROLINE WANG and STEVE PURINTON In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 373-380 Aug. 1988

Avail: CASI HC A02/MF A04

A methodology for automatic mathematical modeling and generating simulation models is described. The models will be verified by running in a test environment using standard profiles with the results compared against known results. The major objective is to create a user friendly environment for engineers to design, maintain, and verify their model and also automatically convert the mathematical model into conventional code for conventional computation. A demonstration program was designed for modeling the Space Shuttle Main Engine Simulation. It is written in LISP and MACSYMA and runs on a Symbolic 3670 Lisp Machine. The program provides a very friendly and well organized environment for engineers to build a knowledge base for base equations and general information. It contains an initial set of component process elements for the Space Shuttle Main Engine Simulation and a questionnaire that allows the engineer to answer a set of questions to specify a particular model. The system is then able to automatically generate the model and FORTRAN code. The future goal which is under construction is to download the FORTRAN code to VAX/VMS system for conventional computation. The SSME mathematical model will be verified in a test environment and the solution compared with the real data profile. The use of artificial intelligence techniques has shown that the process of the simulation modeling can be simplified.

Author

N88-30357*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

THE SPACE STATION ASSEMBLY PHASE: SYSTEM DESIGN TRADE-OFFS FOR THE FLIGHT TELEROBOTIC SERVICER

JEFFREY H. SMITH, MAX GYAMFI, KENT VOLKMER, and WAYNE ZIMMERMAN In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 381-396 Aug. 1988
(Contract NAS7-918)

Avail: CASI HC A03/MF A04

The effects of a recent study aimed at identifying key issues and trade-offs associated with using a Flight Telerobotic Servicer (FTS) to aid in Space Station assembly-phase tasks is described. The use of automation and robotic (A and R)

technologies for large space systems often involves a substitution of automation capabilities for human EVA or IVA activities. A methodology is presented that incorporates assessment of candidate assembly-phase tasks, telerobotic performance capabilities, development costs, and effects of operational constraints. Changes in the region of cost-effectiveness are examined under a variety of system design assumptions. A discussion of issues is presented with focus on three roles the FTS might serve: as a research-oriented test bed to learn more about space usage of telerobotics; as a research based test bed having an experimental demonstration orientation with limited assembly and servicing applications; or as an operational system to augment EVA and to aid construction of the Space Station and to reduce the program (schedule) risk by increasing the flexibility of mission operations.

Author

N88-30358*# Ford Aerospace and Communications Corp., Seabrook, MD.

A SIMULATION ENGINE: COMBINING AN EXPERT SYSTEM WITH A SIMULATION LANGUAGE

JAMES R. SPIEGEL and DAVID B. LAVALLEE In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 397-406 Aug. 1988

Avail: CASI HC A02/MF A04

Expert systems were applied in a number of ways to the field of simulation. One of these is the application of an expert system to drive a simulation, by making run-time decisions which effect the simulation. This approach was successful for a number of specific simulation models. The simulation engine extends this capability by supporting this type of interaction in a general purpose setting. A general purpose simulation language is provided for building models, and an inference engine is provided for knowledge processing. This combination results in a mechanism which allows general purpose models to be simulated in concert with interactive knowledge bases.

Author

N88-30359*# Science Applications Research, Greenbelt, MD. National Space Science Data Center.

THE ADVICE TAKER/INQUIRER: A SYSTEM FOR HIGH-LEVEL ACQUISITION OF EXPERT KNOWLEDGE

ROBERT F. CROMP In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 409-423 Aug. 1988

Avail: CASI HC A03/MF A04

The Advice Taker/Inquirer (AT/I) is a domain-independent program that is used to construct, monitor, and improve an expert system. In the learning phase, an expert teaches a strategy to the AT/I by providing it with declarative and procedural knowledge, expressed in the expert's domain-specific vocabulary. The expert can modify any advice given to the system earlier, and any advice dependent on the altered advice is reviewed automatically for syntactic and semantic soundness. Knowledge acquisition and methods for ensuring the integrity of the knowledge base in an expert system is discussed.

Author

N88-30360*# Space Telescope Science Inst., Baltimore, MD.
LISP OBJECT STATE SAVER (LOSS): A FACILITY USED TO SAVE PARTIAL SCHEDULES OF THE HUBBLE SPACE TELESCOPE

JEFFREY L. SPONSLER In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of

Artificial Intelligence p 425-433 Aug. 1988

Avail: CASI HC A02/MF A04

Current research in the area of long term scheduling of the Hubble Space Telescope is being done using Common Lisp and Flavors on Lisp Machines. The planning tools manipulate memory-resident data structures which represent the many entities and relationships that represent planning states. The Lisp Object State Saver (LOSS), a general purpose utility, was constructed which allows one to take a snapshot of memory by storing a representation of the structures in a text file. This text file can later be loaded thus restoring the pre-existing and logically equivalent planning state. A LOSS template must be created for each datatype to be stored and a simple grammar governs the creation of such templates.

Author

N88-30361*# Space Telescope Science Inst., Baltimore, MD.

VERIFICATION AND VALIDATION OF RULEBASED SYSTEMS FOR HUBBLE SPACE TELESCOPE GROUND SUPPORT

SHON VICK (Space Telescope Science Inst., Baltimore, MD.) and KELLY LINDENMAYER (Computer Sciences Corp., Baltimore, Md.) In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 435-448 Aug. 1988

Avail: CASI HC A03/MF A04

As rulebase systems become more widely used in operational environments, the focus is on the problems and concerns of maintaining expert systems. In the conventional software model, the verification and validation of a system have two separate and distinct meanings. To validate a system means to demonstrate that the system does what is advertised. The verification process refers to investigating the actual code to identify inconsistencies and redundancies within the logic path. In current literature regarding maintaining rulebased systems, little distinction is made between these two terms. In fact, often the two terms are used interchangeably. Verification and validation of rulebased systems are discussed as separate but equally important aspects of the maintenance phase. Also described are some of the tools and methods that were developed at the Space Telescope Science Institute to aid in the maintenance of the rulebased system.

Author

N88-30362*# Computer Sciences Corp., Beltsville, MD.
System Sciences Div.

THE NEED FOR A COMPREHENSIVE EXPERT SYSTEM DEVELOPMENT METHODOLOGY

JOHN BAUMERT, ANNA CRITCHFIELD, and KAREN LEAVITT In NASA, Goddard Space Flight Center, The 1988 Goddard Conference on Space Applications of Artificial Intelligence p 449-460 Aug. 1988

Avail: CASI HC A03/MF A04

In a traditional software development environment, the introduction of standardized approaches has led to higher quality, maintainable products on the technical side and greater visibility into the status of the effort on the management side. This study examined expert system development to determine whether it differed enough from traditional systems to warrant a reevaluation of current software development methodologies. Its purpose was to identify areas of similarity with traditional software development and areas requiring tailoring to the unique needs of expert systems. A second purpose was to determine whether existing expert system development methodologies meet the needs of expert system development, management, and maintenance personnel. The study consisted of a literature search

and personal interviews. It was determined that existing methodologies and approaches to developing expert systems are not comprehensive nor are they easily applied, especially to cradle to grave system development. As a result, requirements were derived for an expert system development methodology and an initial annotated outline derived for such a methodology.

Author

N89-10063*# National Aeronautics and Space Administration.
Goddard Space Flight Center, Greenbelt, MD.

PROCEEDINGS OF 1987 GODDARD CONFERENCE ON SPACE APPLICATIONS OF ARTIFICIAL INTELLIGENCE (AI) AND ROBOTICS

ELLEN G. STOLARIK, ed., RONALD G. LITTLEFIELD, ed., and DAVID S. BEYER, ed. 1987 718 p Conference held in Greenbelt, Md., 13-14 May 1987
(NASA-TM-89663; NAS 1.15:89663) Avail: CASI HC A99/MF A06

Topics addressed include: planning/scheduling expert systems; fault isolation/diagnosis expert systems; data processing/analysis expert systems; expert system tools/techniques; and robotics.

N89-10064*# National Aeronautics and Space Administration.
Ames Research Center, Moffett Field, CA.

MAINTAINING AN EXPERT SYSTEM FOR THE HUBBLE SPACE TELESCOPE GROUND SUPPORT

KELLY LINDENMAYER (Computer Sciences Corp., Baltimore, MD.), SHON VICK (Space Telescope Science Inst., Baltimore, Md.), and DON ROSENTHAL In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics p 1-13 1987

Avail: CASI HC A03/MF A06

The transformation portion of the Hubble Space Telescope (HST) Proposal Entry Processor System converts astronomer-oriented description of a scientific observing program into a detailed description of the parameters needed for planning and scheduling. The transformation system is one of a very few rulebased expert systems that has ever entered an operational phase. The day to day operations of the system and its rulebase are no longer the responsibility of the original developer. As a result, software engineering properties of the rulebased approach become more important. Maintenance issues associated with the coupling of rules within a rulebased system are discussed and a method is offered for partitioning a rulebase so that the amount of knowledge needed to modify the rulebase is minimized. This method is also used to develop a measure of the coupling strength of the rulebase.

Author

N89-10065*# Bendix Field Engineering Corp., Lanham, MD.
AN EXPERT SYSTEM FOR SCHEDULING REQUESTS FOR COMMUNICATIONS LINKS BETWEEN TDRSS AND ERBS

DAVID R. MCLEAN, RONALD G. LITTLEFIELD, and DAVID S. BEYER In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 16 p 1987

Avail: CASI HC A03/MF A06

An ERBS-TDRSS Contact Planning System (ERBS-TDRSS CPS) is described which uses a graphics interface and the NASA Transportable Interference Engine. The procedure involves transfer of the ERBS-TDRSS Ground Track Orbit Prediction data to the ERBS flight operations area, where

the ERBS-TDRSS CPS automatically generates requests for TDRSS service. As requested events are rejected, alternative context sensitive strategies are employed to generate new requested events until a schedule is completed. A report generator builds schedule requests for separate ERBS-TDRSS contacts.

IAA

N89-10066*# Ford Aerospace and Communications Corp., College Park, MD. Space Station Programs.

THE MISSION OPERATIONS PLANNING ASSISTANT
JAMES G. SCHUETZLE In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 16 p 1987

Avail: CASI HC A03/MF A06

The Mission Operations Planning Assistant (MOPA) is a knowledge-based system developed to support the planning and scheduling of instrument activities on the Upper Atmospheric Research Satellite (UARS). The MOPA system represents and maintains instrument plans at two levels of abstraction in order to keep plans comprehensible to both UARS Principal Investigators and Command Management personnel. The hierarchical representation of plans also allows MOPA to automatically create detailed instrument activity plans from which spacecraft command loads may be generated. The MOPA system was developed on a Symbolics 3640 computer using the ZETALISP and ART languages. MOPA's features include a textual and graphical interface for plan inspection and modification, recognition of instrument operational constraint violations during the planning process, and consistency maintenance between the different planning levels. This paper describes the current MOPA system.

IAA

N89-10067*# National Aeronautics and Space Administration, Langley Research Center, Hampton, VA.

AUTOPLAN: A PC-BASED AUTOMATED MISSION PLANNING TOOL

FRANK C. PATERRA (National Aeronautics and Space Administration, Langley Research Center, Hampton, VA.), MARC S. ALLEN (Computer Technology Associates, Inc., Newport News, Va.), and GEORGE F. LAWRENCE In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 16 p 1987

Avail: CASI HC A03/MF A06

A PC-based automated mission and resource planning tool, AUTOPLAN, is described, with application to small-scale planning and scheduling systems in the Space Station program. The input is a proposed mission profile, including mission duration, number of allowable slip periods, and requirement profiles for one or more resources as a function of time. A corresponding availability profile is also entered for each resource over the whole time interval under study. AUTOPLAN determines all integrated schedules which do not require more than the available resources.

IAA

N89-10068*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

PLAN-IT: SCHEDULING ASSISTANT FOR SOLAR SYSTEM EXPLORATION

WILLIAM C. DIAS (ITT Federal Electric Corp., Vandenberg AFB, Calif.), JULIA A. HENRICKS, and JENNIFER C. WONG In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial

Intelligence (AI) and Robotics 16 p 1987

Avail: CASI HC A03/MF A06

A frame-based expert scheduling system shell, PLAN-IT, is developed for spacecraft scheduling in the Request Integration Phase, using the Comet Rendezvous Asteroid Flyby (CRAF) mission as a development base. Basic, structured, and expert scheduling techniques are reviewed. Data elements such as activity representation and resource conflict representation are discussed. Resource constraints include minimum and maximum separation times between activities, percentage of time pointed at specific targets, and separation time between targeted intervals of a given activity. The different scheduling technique categories and the rationale for their selection are also considered. IAA

N89-10069*# Contel Federal Systems, Inc., Gaithersburg, MD. **AN EXPERT SYSTEM THAT PERFORMS A SATELLITE STATION KEEPING MANEUVER**

M. KATE LINESBROWNING and JOHN L. STONE, JR. In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 13 p 1987

Avail: CASI HC A03/MF A06

The development and characteristics of a prototype expert system, Expert System for Satellite Orbit Control (ESSOC), capable of providing real-time spacecraft system analysis and command generation for a geostationary satellite are described. The ESSOC recommends appropriate commands that reflect both the changing spacecraft condition and previous procedural action. An internal knowledge base stores satellite status information and is updated with processed spacecraft telemetry. Procedural structure data are encoded in production rules. Structural methods of knowledge acquisition and the design and performance-enhancing techniques that enable ESSOC to operate in real time are also considered.

IAA

N89-10070*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

DEEP SPACE NETWORK RESOURCE SCHEDULING APPROACH AND APPLICATION

WILLIAM C. EGGERMEYER and ALAN BOWLING In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 9 p 1987

Avail: CASI HC A02/MF A06

Deep Space Network (DSN) resource scheduling is the process of distributing ground-based facilities to track multiple spacecraft. The Jet Propulsion Laboratory has carried out extensive research to find ways of automating this process in an effort to reduce time and manpower costs. This paper presents a resource-scheduling system entitled PLAN-IT with a description of its design philosophy. The PLAN-IT's current on-line usage and limitations in scheduling the resources of the DSN are discussed, along with potential enhancements for DSN application.

Author

N89-10071*# Computer Sciences Corp., Beltsville, MD. System Sciences Div.

SPACE STATION PLATFORM MANAGEMENT SYSTEM (PMS) REPLANNING USING RESOURCE ENVELOPES

JOY LEE BUSH, ANNA CRITCHFIELD, and AUDREY LOOMIS In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 17 p 1987

Avail: CASI HC A03/MF A06

One of the responsibilities of the Space Station Platform Management System (PMS) is to maintain constraint-free, short-term plans for platform and free-flyer activities. Both the replanning function and the associated constraint-checking function are viewed as potentially requiring expert system assistance. The PMS Resource Envelope Scheduling System (PRESS) expert system, which is currently under development, is described. The PRESS capabilities will include the following: plan, replan, and perform constraint checking using resource envelopes resembling those required for telescience; initialize itself using the results from a previous run; infer the replanning needs associated with a change in resource availability; allow the user to determine the level of interaction (including an advisory capability) with the system during execution; and generate both a graphic timeline and a report as output. The PRESS is being developed on an IBM PC/AT using TeKnowledge, Inc.'s M.I expert system shell. The PRESS activity definitions and constraints are based on those defined for the Cosmic Background Explorer (COBE) mission scheduled for launch in early 1989.

Author

N89-10072*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

CLEAR: COMMUNICATIONS LINK EXPERT ASSISTANCE RESOURCE

LARRY G. HULL and PETER M. HUGHES In its Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 12 p 1987

Avail: CASI HC A03/MF A06

Communications Link Expert Assistance Resource (CLEAR) is a real time, fault diagnosis expert system for the Cosmic Background Explorer (COBE) Mission Operations Room (MOR). The CLEAR expert system is an operational prototype which assists the MOR operator/analyst by isolating and diagnosing faults in the spacecraft communication link with the Tracking and Data Relay Satellite (TDRS) during periods of realtime data acquisition. The mission domain, user requirements, hardware configuration, expert system concept, tool selection, development approach, and system design were discussed. Development approach and system implementation are emphasized. Also discussed are system architecture, tool selection, operation, and future plans.

Author

N89-10073*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE LOAD SHEDDING ADVISOR: AN EXAMPLE OF A CRISIS-RESPONSE EXPERT SYSTEM

TERRY B. BOLLINGER (Software Productivity Consortium, Reston, VA.), ERIC LIGHTNER (Bendix Field Engineering Corp., Greenbelt, MD.), JOHN LAVERTY (Bendix Field Engineering Corp., Greenbelt, Md.), and EDWARD AMBROSE In its Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 11 p 1987

Avail: CASI HC A03/MF A06

A Prolog-based prototype expert system is described that was implemented by the Network Operations Branch of the NASA Goddard Space Flight Center. The purpose of the prototype was to test whether a small, inexpensive computer system could be used to host a load shedding advisor, a system which would monitor major physical environment parameters in a computer facility, then recommend appropriate operator responses whenever a serious condition was detected. The resulting prototype performed significantly to efficiency

gains achieved by replacing a purely rule-based design methodology with a hybrid approach that combined procedural, entity-relationship, and rule-based methods.

Author

N89-10074*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

A LOCAL AREA COMPUTER NETWORK EXPERT SYSTEM FRAMEWORK

ROBERT DOMINY In its Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 5 p 1987 Prepared in cooperation with Bendix Field Engineering Corp., Greebelt, Md.

Avail: CASI HC A01/MF A06

Over the past years an expert system called LANES designed to detect and isolate faults in the Goddard-wide Hybrid Local Area Computer Network (LACN) was developed. As a result, the need for developing a more generic LACN fault isolation expert system has become apparent. An object oriented approach was explored to create a set of generic classes, objects, rules, and methods that would be necessary to meet this need. The object classes provide a convenient mechanism for separating high level information from low level network specific information. This approach yields a framework which can be applied to different network configurations and be easily expanded to meet new needs.

Author

N89-10075*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

FIESTA: AN OPERATIONAL DECISION AID FOR SPACE NETWORK FAULT ISOLATION

DAWN LOWE (National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.), BOB QUILLIN (Stanford Telecommunications, Inc., McLean, VA.), NADINE MATTESON (Stanford Telecommunications, Inc., McLean, VA.), BILL WILKINSON (Stanford Telecommunications, Inc., McLean, VA.), and STEVE MIKSELL (Stanford Telecommunications, Inc., McLean, Va.) In its Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 21 p 1987 Prepared in cooperation with Bendix Field Engineering Corp., Greenbelt, Md.

(Contract NAS5-27600)

Avail: CASI HC A03/MF A06

The Fault Tolerance Expert System for Tracking and Data Relay Satellite System (TDRSS) Applications (FIESTA) is a fault detection and fault diagnosis expert system being developed as a decision aid to support operations in the Network Control Center (NCC) for NASA's Space Network. The operational objectives which influenced FIESTA development are presented and an overview of the architecture used to achieve these goals are provided. The approach to the knowledge engineering effort and the methodology employed are also presented and illustrated with examples drawn from the FIESTA domain.

Author

N89-10076*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

EXPERT SYSTEM SUPPORT FOR HST OPERATIONS

BRYANT CRUSE (Lockheed Missiles and Space Co., Greenbelt, Md.) and CHARLES WENDE In its Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 11 p 1987

Avail: CASI HC A03/MF A06

An expert system is being developed to support vehicle anomaly diagnosis for the Hubble Space Telescope (HST).

Following a study of safemode entry analyses, a prototype system was developed which reads engineering telemetry formats, and when a safemode event is detected, extracts telemetry from the downlink and writes it into a knowledge base for more detailed analyses. The prototype then summarizes vehicle events (limits exceeded, specific failures). This prototype, the Telemetry Analysis Logic for Operations Support (TALOS) uses the Lockheed Expert System (LES) shell, and includes over 1600 facts, 230 rules, and 27 goals. Although considered a prototype, it is already an operationally useful system. The history leading into the TALOS prototype will be discussed, an overview of the present TALOS system will be presented, and the role of the TALOS system in contingency planning will be delineated.

Author

N89-10077*# Stanford Telecommunications, Inc., McLean, VA.

A HIERARCHICALLY DISTRIBUTED ARCHITECTURE FOR FAULT ISOLATION EXPERT SYSTEMS ON THE SPACE STATION

STEVE MIKSELL and SUE COFFER In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 9 p 1987

(Contract NAS5-29280)

Avail: CASI HC A03/MF A06

The Space Station Axiomatic Fault Isolating Expert Systems (SAFTIES) system deals with the hierarchical distribution of control and knowledge among independent expert systems doing fault isolation and scheduling of Space Station subsystems. On its lower level, fault isolation is performed on individual subsystems. These fault isolation expert systems contain knowledge about the performance requirements of their particular subsystem and corrective procedures which may be involved in response to certain performance errors. They can control the functions of equipment in their system and coordinate system task schedules. On a higher level, the Executive contains knowledge of all resources, task schedules for all systems, and the relative priority of all resources and tasks. The executive can override any subsystem task schedule in order to resolve use conflicts or resolve errors that require resources from multiple subsystems. Interprocessor communication is implemented using the SAFTIES Communications Interface (SCI). The SCI is an application layer protocol which supports the SAFTIES distributed multi-level architecture.

Author

N89-10078*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

AUTOMATION OF SPACECRAFT CONTROL CENTERS

ROBERT DUTILLY In its Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 9 p 1987

Avail: CASI HC A05/MF A06

The objective is to describe the further automation of the Payload Operations Control Centers, specifically the Mission Operations Room, by using a series of expert systems interconnected together. The feasibility of using expert systems in the Mission Operations Room is presently being determined. The expert system under development is called the Communications Link Expert Assistance Resource (CLEAR) project. It is the first control center expert system being designed and implemented at Goddard. It will demonstrate the feasibility and practicality of expert systems in a real-time control center environment. There is a two-fold purpose. First is to briefly

describe the present effort of the CLEAR expert system under development. The second is to describe how a series of interacting expert systems could be developed to almost totally automate the Mission Operations Room within the control center. How these expert systems would be put together and what functions they could perform in the control center is described. These efforts will provide a great deal of applicability toward the automation of the space station.

Author

N89-10079*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

**SPACELAB DATA PROCESSING FACILITY (SLDPF)
QUALITY ASSURANCE EXPERT SYSTEMS
DEVELOPMENT**

ANGELITA C. KELLY (National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.), LISA BASILE (National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.), TROY AMES (National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.), JANICE WATSON (Lockheed Missiles and Space Co., Greenbelt, MD.), and WILLIAM DALLAM (Lockheed Missiles and Space Co., Greenbelt, Md.) In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 23 p 1987

Avail: CASI HC A03/MF A06

Spacelab Data Processing Facility (SLDPF) expert system prototypes were developed to assist in the quality assurance of Spacelab and/or Attached Shuttle Payload (ASP) processed telemetry data. The SLDPF functions include the capturing, quality monitoring, processing, accounting, and forwarding of mission data to various user facilities. Prototypes for the two SLDPF functional elements, the Spacelab Output Processing System and the Spacelab Input Processing Element, are described. The prototypes have produced beneficial results including an increase in analyst productivity, a decrease in the burden of tedious analyses, the consistent evaluation of data, and the providing of concise historical records.

IAA

N89-10080*# Space Telescope Science Inst., Baltimore, MD.
AN EXPERT SYSTEM APPROACH TO ASTRO-NOMICAL DATA ANALYSIS

MARK D. JOHNSTON In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 17 p 1987

Avail: CASI HC A03/MF A06

Expert systems technology has much to offer to the problem of astronomical data analysis, where large data volumes and sophisticated analysis goals have caused a variety of interesting problems to arise. The construction of a prototype expert system whose target domain is CCD image calibration, is reported. The prototype is designed to be extensible to different and more complex problems in a straightforward way, and to be largely independent of the details of the specific data analysis system which executes the plan it generates.

Author

N89-10081*# National Aeronautics and Space Administration. Ames Research Center, Moffett Field, CA.

EXPERT SYSTEMS TOOLS FOR HUBBLE SPACE TELESCOPE OBSERVATION SCHEDULING

GLENN MILLER (Computer Sciences Corp., Baltimore, MD.), DON ROSENTHAL (National Aeronautics and Space Administration. Ames Research Center, Moffett Field, CA.),

WILLIAM COHEN (Computer Sciences Corp., Baltimore, MD.), and MARK JOHNSTON (Space Telescope Science Inst., Baltimore, Md.) In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 13 p 1987

Avail: CASI HC A03/MF A06

The utility of expert systems techniques for the Hubble Space Telescope (HST) planning and scheduling is discussed and a plan for development of expert system tools which will augment the existing ground system is described. Additional capabilities provided by these tools will include graphics-oriented plan evaluation, long-range analysis of the observation pool, analysis of optimal scheduling time intervals, constructing sequences of spacecraft activities which minimize operational overhead, and optimization of linkages between observations. Initial prototyping of a scheduler used the Automated Reasoning Tool running on a LISP workstation.

IAA

N89-10082*# George Washington Univ., Washington, DC. Inst. for Artificial Intelligence.

TOWARD AN EXPERT PROJECT MANAGEMENT SYSTEM

BARRY G. SILVERMAN (George Washington Univ., Washington, DC.), ARTHUR MURRAY (George Washington Univ., Washington, DC.), COTY DIAKITE (Intellitek, Inc., Rockville, Md.), and KOSTAS FEGGOS In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 18 p 1987

Avail: CASI HC A03/MF A06

The purpose of the research effort is to prescribe a generic reusable shell that any project office can install and customize for the purposes of advising, guiding, and supporting project managers in that office. The prescribed shell is intended to provide both: a component that generates prescriptive guidance for project planning and monitoring activities, and an analogy (intuition) component that generates descriptive insights of previous experience of successful project managers. The latter component is especially significant in that it has the potential to: retrieve insights, not just data, and provide a vehicle for expert PMs to easily transcribe their current experiences in the course of each new project managed.

Author

N89-10083*# Computer Sciences Corp., Baltimore, MD. Astronomy Programs.

A NATURAL LANGUAGE QUERY SYSTEM FOR HUBBLE SPACE TELESCOPE PROPOSAL SELECTION

THOMAS HORNICK, WILLIAM COHEN, and GLENN MILLER In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 17 p 1987 Prepared in cooperation with Space Telescope Science Inst., Baltimore, Md.

Avail: CASI HC A03/MF A06

The proposal selection process for the Hubble Space Telescope is assisted by a robust and easy to use query program (TACOS). The system parses an English subset language sentence regardless of the order of the keyword phases, allowing the user a greater flexibility than a standard command query language. Capabilities for macro and procedure definition are also integrated. The system was designed for flexibility in both use and maintenance. In addition, TACOS can be applied to any knowledge domain that can be expressed in terms of a single reaction. The system was implemented mostly in Common LISP. The TACOS design is described in detail, with particular

attention given to the implementation methods of sentence processing.

Author

N89-10084*# Ford Aerospace and Communications Corp., College Park, MD. Space Missions Div.

MAINTAINING CONSISTENCY BETWEEN PLANNING HIERARCHIES: TECHNIQUES AND APPLICATIONS

DAVID R. ZOCH In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 18 p 1987

Avail: CASI HC A03/MF A06

In many planning and scheduling environments, it is desirable to be able to view and manipulate plans at different levels of abstraction, allowing the users the option of viewing and manipulating either a very detailed representation of the plan or a high-level more abstract version of the plan. Generating a detailed plan from a more abstract plan requires domain-specific planning/scheduling knowledge; the reverse process of generating a high-level plan from a detailed plan Reverse Plan Maintenance, or RPM) requires having the system remember the actions it took based on its domain-specific knowledge and its reasons for taking those actions. This reverse plan maintenance process is described as implemented in a specific planning and scheduling tool, The Mission Operations Planning Assistant (MOPA), as well as the applications of RPM to other planning and scheduling problems; emphasizing the knowledge that is needed to maintain the correspondence between the different hierarchical planning levels.

Author

N89-10085*# Ford Aerospace and Communications Corp., College Park, MD.

A LISP-ADA CONNECTION

ALLAN JAWORSKI, DAVID LAVALLEE, and DAVID ZOCH In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 14 p 1987

Avail: CASI HC A03/MF A06

The prototype demonstrates the feasibility of using Ada for expert systems and the implementation of an expert-friendly interface which supports knowledge entry. In the Ford LISP-Ada Connection (FLAC) system LISP and Ada are used in ways which complement their respective capabilities. Future investigation will concentrate on the enhancement of the expert knowledge entry/debugging interface and on the issues associated with multitasking and real-time expert systems implementation in Ada.

Author

N89-10086*# Computer Sciences Corp., Baltimore, MD. Astronomy Programs.

EXPERT SYSTEMS BUILT BY THE EXPERT: AN EVALUATION OF OPS5

ROBERT JACKSON In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 14 p 1987 Prepared in cooperation with Space Telescope Science Inst., Baltimore, Md.

Avail: CASI HC A03/MF A06

Two expert systems were written in OPS5 by the expert, a Ph.D. astronomer with no prior experience in artificial intelligence or expert systems, without the use of a knowledge engineer. The first system was built from scratch and uses 146 rules to check for duplication of scientific information within a pool of prospective observations. The second system was grafted onto another expert system and uses 149 additional rules to

estimate the spacecraft and ground resources consumed by a set of prospective observations. The small vocabulary, the IF this occurs THEN do that logical structure of OPS5, and the ability to follow program execution allowed the expert to design and implement these systems with only the data structures and rules of another OPS5 system as an example. The modularity of the rules in OPS5 allowed the second system to modify the rulebase of the system onto which it was grafted without changing the code or the operation of that system. These experiences show that experts are able to develop their own expert systems due to the ease of programming and code reusability in OPS5. Author

N89-10087*# National Aeronautics and Space Administration. Langley Research Center, Hampton, VA.

SPACE TRUSS ASSEMBLY USING TELEOPERATED MANIPULATORS

WALTER W. HANKINS, III (National Aeronautics and Space Administration. Langley Research Center, Hampton, VA.), RANDOLPH W. MIXON (National Aeronautics and Space Administration. Langley Research Center, Hampton, VA.), HOWARD C. JONES (National Aeronautics and Space Administration. Langley Research Center, Hampton, VA.), and THOMAS W. BURGESS (Oak Ridge National Lab., Tenn.) In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 18 p 1987

Avail: CASI HC A03/MF A06

Teleoperator experiments were conducted which have demonstrated that a realistic, complex task, typical of those accomplished on-orbit by EVA astronauts, can be done in a smooth, timely manner with manipulators remotely controlled by humans. The real concerns were: (1) do manipulators have sufficient dexterity for these tasks, (2) can sufficient information from the remote site be provided to permit adequate teleoperator control, (3) can reasonable times relative to EVA times be achieved, (4) can the task be completed without frequent and/or damaging impacts among the task components and the manipulators? Positive answers were found to all of these concerns. Tasks times, operator fatigue, and smoothness of operation could be improved by designing the task components and the manipulators for greater compatibility. The data recorded supplements a data base of performance metrics for the same task done in the water immersion training facility as well as space flight and provides management with an objective basis for deciding how and where to apply manipulators in space. Author

N89-10088*# California Univ., Berkeley, CA. Telerobotics Unit.

A UNIVERSITY TEACHING SIMULATION FACILITY

LAWRENCE STARK, WON-SOO KIM, FRANK TENDICK, MITCHELL TYLER, BLAKE HANNAFORD, WISSAM BARAKAT, OLAF BERGENGRUEN, LOUIS BRADDI, JOSEPH EISENBERG, STEPHEN ELLIS et al. In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 30 p 1987

Avail: CASI HC A03/MF A06

An experimental telerobotics (TR) simulation is described suitable for studying human operator (HO) performance. Simple manipulator pick-and-place and tracking tasks allowed quantitative comparison of a number of calligraphic display viewing conditions. A number of control modes could be compared in this TR simulation, including

displacement, rate, and acceleratory control using position and force joysticks. A homeomorphic controller turned out to be no better than joysticks; the adaptive properties of the HO can apparently permit quite good control over a variety of controller configurations and control modes. Training by optimal control example seemed helpful in preliminary experiments. Author

N89-10089*# Grumman Aerospace Corp., Bethpage, NY. **OPEN CONTROL/DISPLAY SYSTEM FOR A TELEROBOTICS WORK STATION**

SAUL KESLOWITZ In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 21 p 1987

Avail: CASI HC A03/MF A06

A working Advanced Space Cockpit was developed that integrated advanced control and display devices into a state-of-the-art multimicroprocessor hardware configuration, using window graphics and running under an object-oriented, multitasking real-time operating system environment. This Open Control/Display System supports the idea that the operator should be able to interactively monitor, select, control, and display information about many payloads aboard the Space Station using sets of I/O devices with a single, software-reconfigurable workstation. This is done while maintaining system consistency, yet the system is completely open to accept new additions and advances in hardware and software. The Advanced Space Cockpit, linked to Grumman's Hybrid Computing Facility and Large Amplitude Space Simulator (LASS), was used to test the Open Control/Display System via full-scale simulation of the following tasks: telerobotic truss assembly, RCS and thermal bus servicing, CMG changeout, RMS constrained motion and space constructible radiator assembly, HPA coordinated control, and OMV docking and tumbling satellite retrieval. The proposed man-machine interface standard discussed has evolved through many iterations of the tasks, and is based on feedback from NASA and Air Force personnel who performed those tasks in the LASS. Author

N89-10090*# Human Machine Interfaces, Inc., Knoxville, TN. Fuel Recycle Div.

CONSOLIDATED FUEL REPROCESSING PROGRAM: THE IMPLICATIONS OF FORCE REFLECTION FOR TELEOPERATION IN SPACE

JOHN V. DRAPER, JOSEPH N. HERNDON, and WENDY E. MOORE In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 16 p 1987 Prepared in cooperation with Martin Marietta Aerospace, Denver, Colo.

(Contract DE-AC05-84OR-21400)

Avail: CASI HC A03/MF A06

Previous research on teleoperator force feedback is reviewed and results of a testing program which assessed the impact of force reflection on teleoperator task performance are reported. Force reflection is a type of force feedback in which the forces acting on the remote portion of the teleoperator are displayed to the operator by back-driving the master controller. The testing program compared three force reflection levels: 4 to 1 (four units of force on the slave produce one unit of force at the master controller), 1 to 1, and infinity to 1 (no force reflection). Time required to complete tasks, rate of occurrence of errors, the maximum force applied to tasks components, and variability in forces applied to components during completion of representative remote handling tasks were used as dependent variables. Operators exhibited lower error rates, lower peak

forces, and more consistent application of forces using force reflection than they did without it. These data support the hypothesis that force reflection provides useful information for teleoperator users. The earlier literature and the results of the experiment are discussed in terms of their implications for space based teleoperator systems. The discussion described the impact of force reflection on task completion performance and task strategies, as suggested by the literature. It is important to understand the trade-offs involved in using telerobotic systems with and without force reflection.

Author

N89-10091*# Martin Marietta Aerospace, Denver, CO. ISSUES, CONCERNS, AND INITIAL IMPLEMENTATION RESULTS FOR SPACE BASED TELEROBOTIC CONTROL

D. A. LAWRENCE, J. D. CHAPEL, and T. M. DEPKOVICH In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 18 p 1987

Avail: CASI HC A03/MF A06

Telerobotic control for space based assembly and servicing tasks presents many problems in system design. Traditional force reflection teleoperation schemes are not well suited to this application, and the approaches to compliance control via computer algorithms have yet to see significant testing and comparison. These observations are discussed in detail, as well as the concerns they raise for imminent design and testing of space robotic systems. As an example of the detailed technical work yet to be done before such systems can be specified, a particular approach to providing manipulator compliance is examined experimentally and through modeling and analysis. This yields some initial insight into the limitations and design trade-offs for this class of manipulator control schemes. Implications of this investigation for space based telerobots are discussed in detail.

Author

N89-10092*# RCA Advanced Technology Labs., Moorestown, NJ.

A SHARED POSITION/FORCE CONTROL METHODOLOGY FOR TELEOPERATION

JIN S. LEE In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 12 p 1987

Avail: CASI HC A03/MF A06

A flexible and computationally efficient shared position/force control concept and its implementation in the Robot Control C Library (RCCL) are presented from the point of teleoperation. This methodology enables certain degrees of freedom to be position-controlled through real time manual inputs and the remaining degrees of freedom to be force-controlled by computer. Functionally, it is a hybrid control scheme in that certain degrees of freedom are designated to be under position control, and the remaining degrees of freedom to be under force control. However, the methodology is also a shared control scheme because some degrees of freedom can be put under manual control and the other degrees of freedom put under computer control. Unlike other hybrid control schemes, which process position and force commands independently, this scheme provides a force control loop built on top of a position control inner loop. This feature minimizes the computational burden and increases disturbance rejection. A simple implementation is achieved partly because the joint control servos that are part of most robots can be used to provide the position control inner loop. Along with this control scheme,

several menus were implemented for the convenience of the user. The implemented control scheme was successfully demonstrated for the tasks of hinged-panel opening and peg-in-hole insertion.

Author

N89-10093*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

MULTIPLE SENSOR SMART ROBOT HAND WITH FORCE CONTROL

RICHARD R. KILLION, LEE R. ROBINSON, and ANTAL BEJCZY In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 20 p 1987

Avail: CASI HC A03/MF A06

A smart robot hand developed at JPL for the Protoflight Manipulator Arm (PFMA) is described. The development of this smart hand was based on an integrated design and subsystem architecture by considering mechanism, electronics, sensing, control, display, and operator interface in an integrated design approach. The mechanical details of this smart hand and the overall subsystem are described elsewhere. The sensing and electronics components of the JPL/PFMA smart hand are summarized and it is described in some detail in control capabilities.

Author

N89-10094*# McDonnell-Douglas Astronautics Co., Houston, TX. Engineering Services.

AN OPTIMAL RESOLVED RATE LAW FOR KINEMATICALLY REDUNDANT MANIPULATORS

B. J. BOURGEOIS In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 20 p 1987

Avail: CASI HC A03/MF A06

The resolved rate law for a manipulator provides the instantaneous joint rates required to satisfy a given instantaneous hand motion. When the joint space has more degrees of freedom than the task space, the manipulator is kinematically redundant and the kinematic rate equations are underdetermined. These equations can be locally optimized, but the resulting pseudo-inverse solution was found to cause large joint rates in some cases. A weighting matrix in the locally optimized (pseudo-inverse) solution is dynamically adjusted to control the joint motion as desired. Joint reach limit avoidance is demonstrated in a kinematically redundant planar arm model. The treatment is applicable to redundant manipulators with any number of revolute joints and to nonplanar manipulators.

Author

N89-10095*# Maryland Univ., College Park, MD. Robotics Lab.

AN ADAPTIVE CONTROL SCHEME FOR A FLEXIBLE MANIPULATOR

T. C. YANG (Maryland Univ., College Park.), J. C. S. YANG (Maryland Univ., College Park.), and P. KUDVA (Advanced Technology and Research, Inc., Burtonsville, Md.) In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 14 p 1987

Avail: CASI HC A03/MF A06

The problem of controlling a single link flexible manipulator is considered. A self-tuning adaptive control scheme is proposed which consists of a least squares on-line parameter identification of an equivalent linear model followed by a tuning of the gains of a pole placement controller using the parameter estimates. Since the initial parameter values for this model are

assumed unknown, the use of arbitrarily chosen initial parameter estimates in the adaptive controller would result in undesirable transient effects. Hence, the initial stage control is carried out with a PID controller. Once the identified parameters have converged, control is transferred to the adaptive controller. Naturally, the relevant issues in this scheme are tests for parameter convergence and minimization of overshoots during control switch-over. To demonstrate the effectiveness of the proposed scheme, simulation results are presented with an analytical nonlinear dynamic model of a single link flexible manipulator.

Author

N89-10096*# National Aeronautics and Space Administration. Ames Research Center, Moffett Field, CA.

ADVANCED DATA MANAGEMENT DESIGN FOR AUTONOMOUS TELEROBOTIC SYSTEMS IN SPACE USING SPACEBORNE SYMBOLIC PROCESSORS

ANDRE GOFORTH In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 19 p 1987

Avail: CASI HC A03/MF A06

The use of computers in autonomous telerobots is reaching the point where advanced distributed processing concepts and techniques are needed to support the functioning of Space Station era telerobotic systems. Three major issues that have impact on the design of data management functions in a telerobot are covered. It also presents a design concept that incorporates an intelligent systems manager (ISM) running on a spaceborne symbolic processor (SSP), to address these issues. The first issue is the support of a system-wide control architecture or control philosophy. Salient features of two candidates are presented that impose constraints on data management design. The second issue is the role of data management in terms of system integration. This refers to providing shared or coordinated data processing and storage resources to a variety of telerobotic components such as vision, mechanical sensing, real-time coordinated multiple limb and end effector control, and planning and reasoning. The third issue is hardware that supports symbolic processing in conjunction with standard data I/O and numeric processing. A SSP that currently is seen to be technologically feasible and is being developed is described and used as a baseline in the design concept. Author

N89-10097*# Little (Arthur D.), Inc., Cambridge, MA.

ROBOT HANDS AND EXTRAVEHICULAR ACTIVITY

BETH MARCUS In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 16 p 1987

Avail: CASI HC A03/MF A06

Extravehicular activity (EVA) is crucial to the success of both current and future space operations. As space operations have evolved in complexity so has the demand placed on the EVA crewman. In addition, some NASA requirements for human capabilities at remote or hazardous sites were identified. One of the keys to performing useful EVA tasks is the ability to manipulate objects accurately, quickly and without early or excessive fatigue. The current suit employs a glove which enables the crewman to perform grasping tasks, use tools, turn switches, and perform other tasks for short periods of time. However, the glove's bulk and resistance to motion ultimately causes fatigue. Due to this limitation it may not be possible to meet the productivity requirements that will be placed on the EVA crewman of the future with the current or developmental Extravehicular Mobility Unit (EMU) hardware. In addition, this

hardware will not meet the requirements for remote or hazardous operations. In an effort to develop ways for improving crew productivity, a contract was awarded to develop a prototype anthropomorphic robotic hand (ARH) for use with an extravehicular space suit. The first step in this program was to perform a design study which investigated the basic technology required for the development of an ARH to enhance crew performance and productivity. The design study phase of the contract and some additional development work is summarized.

Author

N89-10098*# Oak Ridge National Lab., TN. Center for Engineering Systems Advanced Research.

DYNAMIC TASK ALLOCATION FOR A MAN-MACHINE SYMBIOTIC SYSTEM

L. E. PARKER and F. G. PIN In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 19 p 1987 Submitted for publication

(Contract DE-AC05-84OR-21400)

Avail: CASI HC A03/MF A06

This report presents a methodological approach to the dynamic allocation of tasks in a man-machine symbiotic system in the context of dexterous manipulation and teleoperation. This report addresses a symbiotic system containing two symbiotic partners which work toward controlling a single manipulator arm for the execution of a series of sequential manipulation tasks. It is proposed that an automated task allocator use knowledge about the constraints/criteria of the problem, the available resources, the tasks to be performed, and the environment to dynamically allocate task recommendations for the man and the machine. The presentation of the methodology includes discussions concerning the interaction of the knowledge areas, the flow of control, the necessary communication links, and the replanning of the task allocation. Examples of task allocation are presented to illustrate the results of this methodology.

DOE

N89-10099*# National Bureau of Standards, Gaithersburg, MD. Robot Systems Div.

NASREN: STANDARD REFERENCE MODEL FOR TELEROBOT CONTROL

J. S. ALBUS, R. LUMIA, and H. MCCAIN In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 19 p 1987

Avail: CASI HC A03/MF A06

A hierarchical architecture is described which supports space station telerobots in a variety of modes. The system is divided into three hierarchies: task decomposition, world model, and sensory processing. Goals at each level of the task dedomposition heirarchy are divided both spatially and temporally into simpler commands for the next lower level. This decomposition is repreated until, at the lowest level, the drive signals to the robot actuators are generated. To accomplish its goals, task decomposition modules must often use information stored in the world model. The purpose of the sensory system is to update the world model as rapidly as possible to keep the model in registration with the physical world. The architecture of the entire control system hierach is described and how it can be applied to space telerobot applications.

Author

N89-10100*# RCA Astro-Electronics Div., Princeton, NJ. Space Div.

KINEMATIC STUDY OF FLIGHT TELEROBOTIC

SERVICER CONFIGURATION ISSUES

R. H. LEWIS, R. D. SCOTT, and W. S. HOWARD In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 17 p 1987

Avail: CASI HC A03/MF A06

Several factors, such as body size and shape, and the number of arms and their placement, will influence how well the Flight Telerobotic Servicer (FTS) is suited to its potential duties for the Space Station Program. In order to examine the implications of these configuration options, eight specific 2, 3, and 4 armed FTS configuration were simulated and used to perform a Space Station Orbital Replacement Unit (ORU) exchange. The strengths and weaknesses of each configuration were evaluated. Although most of the configurations examined were able to perform the exchange, several of the 3 and 4 arm configurations had operational advantages. The results obtained from these simulations are specific to the assumptions associated with the ORU exchange scenario examined. However, they do illustrate the general interrelationships and sensitivities which need to be understood.

Author

N89-10101*# Martin Marietta Aerospace, Denver, CO. Advance Automation Technology (Robotics) Group.

ACTUATORS FOR A SPACE MANIPULATOR

W. CHUN and P. BRUNSON In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 20 p 1987

Avail: CASI HC A03/MF A06

The robotic manipulator can be decomposed into distinct subsystems. One particular area of interest of mechanical subsystems is electromechanical actuators (or drives). A drive is defined as a motor with an appropriate transmission. An overview is given of existing, as well as state-of-the-art drive systems. The scope is limited to space applications. A design philosophy and adequate requirements are the initial steps in designing a space-qualified actuator. The focus is on the d-c motor in conjunction with several types of transmissions (harmonic, tendon, traction, and gear systems). The various transmissions will be evaluated and key performance parameters will be addressed in detail. Included in the assessment is a shuttle RMS joint and a MSFC drive of the Prototype Manipulator Arm. Compound joints are also investigated. Space imposes a set of requirements for designing a high-performance drive assembly. Its inaccessibility and cryogenic conditions warrant special considerations. Some guidelines concerning these conditions are present. The goal is to gain a better understanding in designing a space actuator.

Author

N89-10102*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

CABLE APPLICATIONS IN ROBOT COMPLIANT DEVICES

JAMES J. KERLEY In its Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 11 p 1987

Avail: CASI HC A03/MF A06

Robotic systems need compliance to connect the robot to the work object. The cable system illustrated offers compliance for mating but can be changed in space to become quite stiff. Thus the same system can do both tasks, even in environments where the work object or robot are moving at different frequencies and different amplitudes. The adjustment

can be made in all six degrees of freedom, translated in or rotated in any plane and still make a good contact and control. Author

N89-10103*# National Aeronautics and Space Administration. John F. Kennedy Space Center, Cocoa Beach, FL.

COMPUTER HARDWARE AND SOFTWARE FOR ROBOTIC CONTROL

VIRGIL LEON DAVIS In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 19 p 1987

Avail: CASI HC A03/MF A06

The KSC has implemented an integrated system that coordinates state-of-the-art robotic subsystems. It is a sensor based real-time robotic control system performing operations beyond the capability of an off-the-shelf robot. The integrated system provides real-time closed loop adaptive path control of position and orientation of all six axes of a large robot; enables the implementation of a highly configurable, expandable testbed for sensor system development; and makes several smart distributed control subsystems (robot arm controller, process controller, graphics display, and vision tracking) appear as intelligent peripherals to a supervisory computer coordinating the overall systems.

Author

N89-10104*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

TELEOPERATED POSITION CONTROL OF A PUMA ROBOT

EDMUND AUSTIN and CHUNG P. FONG In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 19 p 1987

Avail: CASI HC A03/MF A06

A laboratory distributed computer control teleoperator system is developed to support NASA's future space telerobotic operation. This teleoperator system uses a universal force-reflecting hand controller in the local site as the operator's input device. In the remote site, a PUMA controller receives the Cartesian position commands and implements PID control laws to position the PUMA robot. The local site uses two microprocessors while the remote site uses three. The processors communicate with each other through shared memory. The PUMA robot controller was interfaced through custom made electronics to bypass VAL. The development status of this teleoperator system is reported. The execution time of each processor is analyzed, and the overall system throughput rate is reported. Methods to improve the efficiency and performance are discussed.

Author

N89-10105*# Advanced Technology and Research, Inc., Burtonsville, MD.

PERFORMANCE IMPROVEMENT OF ROBOTS USING A LEARNING CONTROL SCHEME

RAMUHALLI KRISHNA (Advanced Technology and Research, Inc., Burtonsville, MD.), PEN-TAI CHIANG (Maryland Univ., College Park.), and JACKSON C. S. YANG (Maryland Univ., College Park.) In NASA, Goddard Space Flight Center, Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics 20 p 1987

Avail: CASI HC A03/MF A06

Many applications of robots require that the same task be repeated a number of times. In such applications, the errors associated with one cycle are also repeated every cycle of the

operation. An off-line learning control scheme is used here to modify the command function which would result in smaller errors in the next operation. The learning scheme is based on a knowledge of the errors and error rates associated with each cycle. Necessary conditions for the iterative scheme to converge to zero errors are derived analytically considering a second order servosystem model. Computer simulations show that the errors are reduced at a faster rate if the error rate is included in the iteration scheme. The results also indicate that the scheme may increase the magnitude of errors if the rate information is not included in the iteration scheme. Modification of the command input using a phase and gain adjustment is also proposed to reduce the errors with one attempt. The scheme is then applied to a computer model of a robot system similar to PUMA 560. Improved performance of the robot is shown by considering various cases of trajectory tracing. The scheme can be successfully used to improve the performance of actual robots within the limitations of the repeatability and noise characteristics of the robot.

Author

N89-26578*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE 1989 GODDARD CONFERENCE ON SPACE APPLICATIONS OF ARTIFICIAL INTELLIGENCE

JAMES RASH, ed. Washington Apr. 1989 385 p Conference held in Greenbelt, MD, 16-17 May 1989
(NASA-CP-3033; REPT-89B00099; NAS 1.55:3033)

Avail: CASI HC A17/MF A04

The following topics are addressed: mission operations support; planning and scheduling; fault isolation/diagnosis; image processing and machine vision; data management; and modeling and simulation.

N89-26579*# National Aeronautics and Space Administration. Ames Research Center, Moffett Field, CA.

KNOWLEDGE BASED AND INTERACTIVE CONTROL FOR THE SUPERFLUID HELIUM ON-ORBIT TRANSFER PROJECT

TIMOTHY P. CASTELLANO (Sterling Software, Palo Alto, CA.), ERIC A. RAYMOND (Sterling Software, Palo Alto, CA.), JEFF C. SHAPIRO (Sterling Software, Palo Alto, CA.), FRANK A. ROBINSON (Sterling Software, Palo Alto, CA.), and DONALD A. ROSENTHAL In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 3-11 Apr. 1989

Avail: CASI HC A02/MF A04

NASA's Superfluid Helium On-Orbit Transfer (SHOOT) project is a Shuttle-based experiment designed to acquire data on the properties of superfluid helium in microgravity. Aft Flight Deck Computer Software for the SHOOT experiment is comprised of several monitoring programs which give the astronaut crew visibility into SHOOT systems and a rule based system which will provide process control, diagnosis and error recovery for a helium transfer without ground intervention. Given present Shuttle manifests, this software will become the first expert system to be used in space. The SHOOT Command and Monitoring System (CMS) software will provide a near real time highly interactive interface for the SHOOT principal investigator to control the experiment and to analyze and display its telemetry. The CMS software is targeted for all phases of the SHOOT project: hardware development, pre-flight pad servicing, in-flight operations, and post-flight data analysis.

Author

N89-26580*# Maryland Univ., College Park, MD. Dept. of Computer Science.

SHARED RESOURCE CONTROL BETWEEN HUMAN AND COMPUTER

JAMES HENDLER and REID WILSON In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 13-19 Apr. 1989 Sponsored by part by ONR, Washington, DC

Avail: CASI HC A02/MF A04

The advantages of an AI system of actively monitoring human control of a shared resource (such as a telerobotic manipulator) are presented. A system is described in which a simple AI planning program gains efficiency by monitoring human actions and recognizing when the actions cause a change in the system's assumed state of the world. This enables the planner to recognize when an interaction occurs between human actions and system goals, and allows maintenance of an up-to-date knowledge of the state of the world and thus informs the operator when human action would undo a goal achieved by the system, when an action would render a system goal unachievable, and efficiently replans the establishment of goals after human intervention.

Author

N89-26581*# Unisys Corp., Lanham, MD.

AN ENGLISH LANGUAGE INTERFACE FOR CONSTRAINED DOMAINS

BRENDA J. PAGE In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 21-34 Apr. 1989

Avail: CASI HC A03/MF A04

The Multi-Satellite Operations Control Center (MSOCC) Jargon Interpreter (MJI) demonstrates an English language interface for a constrained domain. A constrained domain is defined as one with a small and well delineated set of actions and objects. The set of actions chosen for the MJI is from the domain of MSOCC Applications Executive (MAE) Systems Test and Operations Language (STOL) directives and contains directives for signing a cathode ray tube (CRT) on or off, calling up or clearing a display page, starting or stopping a procedure, and controlling history recording. The set of objects chosen consists of CRTs, display pages, STOL procedures, and history files. Translation from English sentences to STOL directives is done in two phases. In the first phase, an augmented transition net (ATN) parser and dictionary are used for determining grammatically correct parsings of input sentences. In the second phase, grammatically typed sentences are submitted to a forward-chaining rule-based system for interpretation and translation into equivalent MAE STOL directives. Tests of the MJI show that it is able to translate individual clearly stated sentences into the subset of directives selected for the prototype. This approach to an English language interface may be used for similarly constrained situations by modifying the MJI's dictionary and rules to reflect the change of domain.

Author

N89-26582*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

GROUND DATA SYSTEMS RESOURCE ALLOCATION PROCESS

CAROL A. BERNER, RALPH DURHAM, and NORMAN B. REILLY In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 37-47 Apr. 1989

Avail: CASI HC A03/MF A04

The Ground Data Systems Resource Allocation Process

at the Jet Propulsion Laboratory provides medium- and long-range planning for the use of Deep Space Network and Mission Control and Computing Center resources in support of NASA's deep space missions and Earth-based science. Resources consist of radio antenna complexes and associated data processing and control computer networks. A semi-automated system was developed that allows operations personnel to interactively generate, edit, and revise allocation plans spanning periods of up to ten years (as opposed to only two or three weeks under the manual system) based on the relative merit of mission events. It also enhances scientific data return. A software system known as the Resource Allocation and Planning Helper (RALPH) merges the conventional methods of operations research, rule-based knowledge engineering, and advanced data base structures. RALPH employs a generic, highly modular architecture capable of solving a wide variety of scheduling and resource sequencing problems. The rule-based RALPH system has saved significant labor in resource allocation. Its successful use affirms the importance of establishing and applying event priorities based on scientific merit, and the benefit of continuity in planning provided by knowledge-based engineering. The RALPH system exhibits a strong potential for minimizing development cycles of resource and payload planning systems throughout NASA and the private sector.

Author

N89-26583*# Mitre Corp., McLean, VA.

A SITUATED REASONING ARCHITECTURE FOR SPACE-BASED REPAIR AND REPLACE TASKS

BEN BLOOM, DEBRA MCGRATH, and JIM SANBORN In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 49-59 Apr. 1989

Avail: CASI HC A03/MF A04

Space-based robots need low level control for collision detection and avoidance, short-term load management, fine-grained motion, and other physical tasks. In addition, higher level control is required to focus strategic decision making as missions are assigned and carried out. Reasoning and control must be responsive to ongoing changes in the environment. Research aimed at bridging the gap between high level artificial intelligence (AI) planning techniques and task-level robot programming for telerobotic systems is described. Situated reasoning is incorporated into AI and Robotics systems in order to coordinate a robot's activity within its environment. An integrated system under development in a component maintenance domain is described. It is geared towards replacing worn and/or failed Orbital Replacement Units (ORUs) designed for use aboard NASA's Space Station Freedom based on the collection of components available at a given time. High level control reasons in component space in order to maximize the number operational component-cells over time, while the task-level controls sensors and effectors, detects collisions, and carries out pick and place tasks in physical space. Situated reasoning is used throughout the system to cope with component failures, imperfect information, and unexpected events.

Author

N89-26584*# Maryland Univ., College Park, MD. Dept. of Computer Science.

PARALLEL PLAN EXECUTION WITH SELF-PROCESSING NETWORKS

C. LYNNE DAUTRECHY and JAMES A. REGGIA In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 61-73 Apr. 1989 (Contract NAG1-885; NSF IRI-84-51430)

Avail: CASI HC A03/MF A04

A critical issue for space operations is how to develop and apply advanced automation techniques to reduce the cost and complexity of working in space. In this context, it is important to examine how recent advances in self-processing networks can be applied for planning and scheduling tasks. For this reason, the feasibility of applying self-processing network models to a variety of planning and control problems relevant to spacecraft activities is being explored. Goals are to demonstrate that self-processing methods are applicable to these problems, and that MIRRORS/II, a general purpose software environment for implementing self-processing models, is sufficiently robust to support development of a wide range of application prototypes. Using MIRRORS/II and marker passing modelling techniques, a model of the execution of a Spaceworld plan was implemented. This is a simplified model of the Voyager spacecraft which photographed Jupiter, Saturn, and their satellites. It is shown that plan execution, a task usually solved using traditional artificial intelligence (AI) techniques, can be accomplished using a self-processing network. The fact that self-processing networks were applied to other space-related tasks, in addition to the one discussed here, demonstrates the general applicability of this approach to planning and control problems relevant to spacecraft activities. It is also demonstrated that MIRRORS/II is a powerful environment for the development and evaluation of self-processing systems.

Author

N89-26585*# Centre d'Etudes et de Recherches, Toulouse (France). Groupement d'Intelligence Artificielle.

MISSION SCHEDULING

CHRISTINE GASPIN In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 75-86 Apr. 1989

Avail: CASI HC A03/MF A04

How a neural network can work, compared to a hybrid system based on an operations research and artificial intelligence approach, is investigated through a mission scheduling problem. The characteristic features of each system are discussed.

Author

N89-26586*# AI Atlanta, Inc., Decatur, GA.

ON THE DEVELOPMENT OF A REACTIVE SENSOR-BASED ROBOTIC SYSTEM

HENRY H. HEXMOOR and WILLIAM E. UNDERWOOD, JR. In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 87-100 Apr. 1989

Avail: CASI HC A03/MF A04

Flexible robotic systems for space applications need to use local information to guide their action in uncertain environments where the state of the environment and even the goals may change. They have to be tolerant of unexpected events and robust enough to carry their task to completion. Tactical goals should be modified while maintaining strategic goals. Furthermore, reactive robotic systems need to have a broader view of their environments than sensory-based systems. An architecture and a theory of representation extending the basic cycles of action and perception are described. This scheme allows for dynamic description of the environment and determining purposive and timely action. Applications of this scheme for assembly and repair tasks using a Universal Machine Intelligence RTX robot are being explored, but the ideas are extendable to other domains. The nature of reactivity for sensor-based robotic systems and implementation issues encountered in developing a prototype are discussed.

Author

N89-26587*# Bendix Field Engineering Corp., Seabrook, MD.
Software and Engineering Services.

PST AND PARR: PLAN SPECIFICATION TOOLS AND A PLANNING AND RESOURCE REASONING SHELL FOR USE IN SATELLITE MISSION PLANNING

DAVID MCLEAN and WEN YEN In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 101-112 Apr. 1989
(Contract NAS5-31000; NAS5-27772)

Avail: CASI HC A03/MF A04

Plan Specification Tools (PST) are tools that allow the user to specify satellite mission plans in terms of satellite activities, relevant orbital events, and targets for observation. The output of these tools is a set of knowledge bases and environmental events which can then be used by a Planning And Resource Reasoning (PARR) shell to build a schedule. PARR is a reactive planning shell which is capable of reasoning about actions in the satellite mission planning domain. Each of the PST tools and PARR are described as well as the use of PARR for scheduling computer usage in the multisatellite operations control center at Goddard Space Flight Center. Author

N89-26588*# McDonnell-Douglas Space Systems Co., Cocoa Beach, FL.

AN APPROACH TO KNOWLEDGE ENGINEERING TO SUPPORT KNOWLEDGE-BASED SIMULATION OF PAYLOAD GROUND PROCESSING AT THE KENNEDY SPACE CENTER

SHAWN McMANUS and MICHAEL McDANIEL In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 113-125 Apr. 1989

Avail: CASI HC A03/MF A04

Planning for processing payloads was always difficult and time-consuming. With the advent of Space Station Freedom and its capability to support a myriad of complex payloads, the planning to support this ground processing maze involves thousands of man-hours of often tedious data manipulation. To provide the capability to analyze various processing schedules, an object oriented knowledge-based simulation environment called the Advanced Generic Accomodations Planning Environment (AGAPE) is being developed. Having nearly completed the baseline system, the emphasis in this paper is directed toward rule definition and its relation to model development and simulation. The focus is specifically on the methodologies implemented during knowledge acquisition, analysis, and representation within the AGAPE rule structure. A model is provided to illustrate the concepts presented. The approach demonstrates a framework for AGAPE rule development to assist expert system development. Author

N89-26589*# Ford Aerospace and Communications Corp., Seabrook, MD. Space Systems Engineering Operation.

A HEURISTIC APPROACH TO INCREMENTAL AND REACTIVE SCHEDULING

JIDE B. ODUBIYI and DAVID R. ZOCH In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 127-140 Apr. 1989
Avail: CASI HC A03/MF A04

An heuristic approach to incremental and reactive scheduling is described. Incremental scheduling is the process of modifying an existing schedule if the initial schedule does not meet its stated initial goals. Reactive scheduling occurs in near real-time in response to changes in available resources or the

occurrence of targets of opportunity. Only minor changes are made during both incremental and reactive scheduling because a goal of re-scheduling procedures is to minimally impact the schedule. The described heuristic search techniques, which are employed by the Request Oriented Scheduling Engine (ROSE), a prototype generic scheduler, efficiently approximate the cost of reaching a goal from a given state and effective mechanisms for controlling search. Author

N89-26590*# Office National d'Etudes et de Recherches Aerospatiales, Toulouse (France).

A METHOD FOR INTERACTIVE SATELLITE FAILURE DIAGNOSIS: TOWARDS A CONNECTIONIST SOLUTION

P. BOURRET (Centre d'Etudes et de Recherches, Toulouse (France).) and JAMES A. REGGIA (Maryland Univ., College Park.) In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 143-152 Apr. 1989

Avail: CASI HC A02/MF A04

Various kinds of processes which allow one to make a diagnosis are analyzed. The analyses then focuses on one of these processes used for satellite failure diagnosis. This process consists of sending the satellite instructions about system status alterations: to mask the effects of one possible component failure or to look for additional abnormal measures. A formal model of this process is given. This model is an extension of a previously defined connectionist model which allows computation of ratios between the likelihoods of observed manifestations according to various diagnostic hypotheses. The expected mean value of these likelihood measures for each possible status of the satellite can be computed in a similar way. Therefore, it is possible to select the most appropriate status according to three different purposes: to confirm an hypothesis, to eliminate an hypothesis, or to choose between two hypotheses. Finally, a first connectionist schema of computation of these expected mean values is given. Author

N89-26591*# Computer Sciences Corp., Beltsville, MD.

BCAUS PROJECT DESCRIPTION AND CONSIDERATION OF SEPARATION OF DATA AND CONTROL

JOY L. BUSH and STEVEN J. WEAVER In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 153-160 Apr. 1989
Avail: CASI HC A02/MF A04

The commonly stated truths that data may be segregated from program control in generic expert system shells and that such tools support straightforward knowledge representation were examined. The ideal of separation of data from program control in expert systems is difficult to realize for a variety of reasons. One approach to achieving this goal is to integrate hybrid collections of specialized shells and tools instead of producing custom systems built with a single all purpose expert system tool. Aspects of these issues are examined in the context of a specific diagnostic expert system application, the Backup Control Mode Analysis and Utility System (BCAUS), being developed for the Gamma Ray Observatory (GRO) spacecraft. The project and the knowledge gained in working on the project are described. Author

N89-26592*# Westinghouse Electric Corp., Baltimore, MD.
TRACKING AND DATA RELAY SATELLITE FAULT ISOLATION AND CORRECTION USING PACES:

POWER AND ATTITUDE CONTROL EXPERT SYSTEM
CAROL-LEE ERIKSON (Westinghouse Electric Corp., Baltimore, MD.) and PEGGY HOOKER (McDonnell-Douglas Space Systems Co., Cocoa Beach, FL.) In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 161-168 Apr. 1989
Avail: CASI HC A02/MF A04

The Power and Attitude Control Expert System (PACES) is an object oriented and rule based expert system which provides spacecraft engineers with assistance in isolating and correcting problems within the Power and Attitude Control Subsystems of the Tracking and Data Relay Satellites (TDRS). PACES is designed to act in a consultant role. It will not interface to telemetry data, thus preserving full operator control over spacecraft operations. The spacecraft engineer will input requested information. This information will include telemetry data, action being performed, problem characteristics, spectral characteristics, and judgments of spacecraft functioning. Questions are answered either by clicking on appropriate responses (for text), or entering numeric values. A context sensitive help facility allows access to additional information when the user has difficulty understanding a question or deciding on an answer. The major functionality of PACES is to act as a knowledge rich system which includes block diagrams, text, and graphics, linked using hypermedia techniques. This allows easy movement among pieces of the knowledge. Considerable documentation of the spacecraft Power and Attitude Control Subsystems is embedded within PACES. The development phase of TDRSS expert system technology is intended to provide NASA with the necessary expertise and capability to define requirements, evaluate proposals, and monitor the development progress of a highly competent expert system for NASA's Tracking and Data Relay Satellite Program. Author

N89-26593*# General Electric Co., Houston, TX. Mission Integration Office.

SPACELAB LIFE SCIENCES-1 ELECTRICAL DIAGNOSTIC EXPERT SYSTEM

C. Y. KAO and W. S. MORRIS In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 169-183 Apr. 1989 (Contract NAS9-17884)

Avail: CASI HC A03/MF A04

The Spacelab Life Sciences-1 (SLS-1) Electrical Diagnostic (SLED) expert system is a continuous, real time knowledge-based system to monitor and diagnose electrical system problems in the Spacelab. After fault isolation, the SLED system provides corrective procedures and advice to the ground-based console operator. The SLED system updates its knowledge about the status of Spacelab every 3 seconds. The system supports multiprocessing of malfunctions and allows multiple failures to be handled simultaneously. Information which is readily available via a mouse click includes: general information about the system and each component, the electrical schematics, the recovery procedures of each malfunction, and an explanation of the diagnosis. Author

N89-26594*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA. Artificial Intelligence Group.

SHARP: A MULTI-MISSION AI SYSTEM FOR SPACECRAFT TELEMETRY MONITORING AND DIAGNOSIS

DENISE L. LAWSON and MARK L. JAMES In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on

Space Applications of Artificial Intelligence p 185-200 Apr. 1989

Avail: CASI HC A03/MF A04

The Spacecraft Health Automated Reasoning Prototype (SHARP) is a system designed to demonstrate automated health and status analysis for multi-mission spacecraft and ground data systems operations. Telecommunications link analysis of the Voyager II spacecraft is the initial focus for the SHARP system demonstration which will occur during Voyager's encounter with the planet Neptune in August, 1989, in parallel with real-time Voyager operations. The SHARP system combines conventional computer science methodologies with artificial intelligence techniques to produce an effective method for detecting and analyzing potential spacecraft and ground systems problems. The system performs real-time analysis of spacecraft and other related telemetry, and is also capable of examining data in historical context. A brief introduction is given to the spacecraft and ground systems monitoring process at the Jet Propulsion Laboratory. The current method of operation for monitoring the Voyager Telecommunications subsystem is described, and the difficulties associated with the existing technology are highlighted. The approach taken in the SHARP system to overcome the current limitations is also described, as well as both the conventional and artificial intelligence solutions developed in SHARP.

Author

N89-26595*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

REDEX: THE RANGING EQUIPMENT DIAGNOSTIC EXPERT SYSTEM

EDWARD C. LUZZAK (Computer Sciences Corp., Beltsville, MD.), K. GOPALAKRISHNAN (Computer Sciences Corp., Beltsville, MD.), and DAVID J. ZILLIG In its The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 201-211 Apr. 1989 (Contract NAS5-31500)

Avail: CASI HC A03/MF A04

REDEX, an advanced prototype expert system that diagnoses hardware failures in the Ranging Equipment (RE) at NASA's Ground Network tracking stations is described. REDEX will help the RE technician identify faulty circuit cards or modules that must be replaced, and thereby reduce troubleshooting time. It features a highly graphical user interface that uses color block diagrams and layout diagrams to illustrate the location of a fault. A semantic network knowledge representation technique was used to model the design structure of the RE. A catalog of generic troubleshooting rules was compiled to represent heuristics that are applied in diagnosing electronic equipment. Specific troubleshooting rules were identified to represent additional diagnostic knowledge that is unique to the RE. Over 50 generic and 250 specific troubleshooting rules have been derived. REDEX is implemented in Prolog on an IBM PC AT-compatible workstation. Block diagram graphics displays are color-coded to identify signals that have been monitored or inferred to have nominal values, signals that are out of tolerance, and circuit cards and functions that are diagnosed as faulty. A hypertext-like scheme is used to allow the user to easily navigate through the space of diagrams and tables. Over 50 graphic and tabular displays have been implemented. REDEX is currently being evaluated in a stand-alone mode using simulated RE fault scenarios. It will soon be interfaced to the RE and tested in an online environment. When completed and fielded, REDEX will be a concrete example of the application of expert systems technology to the problem of improving

performance and reducing the lifecycle costs of operating NASA's communications networks in the 1990's. Author

**N89-26596*# Boeing Computer Services Co., Seattle, WA.
DIAGNOSTIC TOLERANCE FOR MISSING SENSOR
DATA**

ETHAN A. SCARL In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 213-220 Apr. 1989

Avail: CASI HC A02/MF A04

For practical automated diagnostic systems to continue functioning after failure, they must not only be able to diagnose sensor failures but also be able to tolerate the absence of data from the faulty sensors. It is shown that conventional (associational) diagnostic methods will have combinatoric problems when trying to isolate faulty sensors, even if they adequately diagnose other components. Moreover, attempts to extend the operation of diagnostic capability past sensor failure will necessarily compound those difficulties. Model-based reasoning offers a structured alternative that has no special problems diagnosing faulty sensors and can operate gracefully when sensor data is missing.

Author

N89-26597*# Maryland Univ., College Park, MD. Computer Vision Lab.

**ROBOT ACTING ON MOVING BODIES (RAMBO):
PRELIMINARY RESULTS**

LARRY S. DAVIS, DANIEL DEMENTHON, THOR BESTUL, SOTIRIOS ZIAVRAS, H. V. SRINIVASAN, MADJU SIDDALINGAIAH, and DAVID HARWOOD In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 223-236 Apr. 1989

(Contract DACA76-88-C-0008)

Avail: CASI HC A03/MF A04

A robot system called RAMBO is being developed. It is equipped with a camera, which, given a sequence of simple tasks, can perform these tasks on a moving object. RAMBO is given a complete geometric model of the object. A low level vision module extracts and groups characteristic features in images of the object. The positions of the object are determined in a sequence of images, and a motion estimate of the object is obtained. This motion estimate is used to plan trajectories of the robot tool to relative locations nearby the object sufficient for achieving the tasks. More specifically, low level vision uses parallel algorithms for image enhancement by symmetric nearest neighbor filtering, edge detection by local gradient operators, and corner extraction by sector filtering. The object pose estimation is a Hough transform method accumulating position hypotheses obtained by matching triples of image features (corners) to triples of model features. To maximize computing speed, the estimate of the position in space of a triple of features is obtained by decomposing its perspective view into a product of rotations and a scaled orthographic projection. This allows the use of 2-D lookup tables at each stage of the decomposition. The position hypotheses for each possible match of model feature triples and image feature triples are calculated in parallel. Trajectory planning combines heuristic and dynamic programming techniques. Then trajectories are created using parametric cubic splines between initial and goal trajectories. All the parallel algorithms run on a Connection Machine CM-2 with 16K processors.

Author

**N89-26598*# National Aeronautics and Space Administration.
Goddard Space Flight Center, Greenbelt, MD.**

**DEVELOPMENT OF AN INTELLIGENT INTERFACE
FOR ADDING SPATIAL OBJECTS TO A KNOWLEDGE-
BASED GEOGRAPHIC INFORMATION SYSTEM**

WILLIAM J. CAMPBELL and CRAIG GOETTSCHE In its The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 239-247 Apr. 1989

Avail: CASI HC A02/MF A04

Earth Scientists lack adequate tools for quantifying complex relationships between existing data layers and studying and modeling the dynamic interactions of these data layers. There is a need for an earth systems tool to manipulate multi-layered, heterogeneous data sets that are spatially indexed, such as sensor imagery and maps, easily and intelligently in a single system. The system can access and manipulate data from multiple sensor sources, maps, and from a learned object hierarchy using an advanced knowledge-based geographical information system. A prototype Knowledge-Based Geographic Information System (KGIS) was recently constructed. Many of the system internals are well developed, but the system lacks an adequate user interface. A methodology is described for developing an intelligent user interface and extending KGIS to interconnect with existing NASA systems, such as imagery from the Land Analysis System (LAS), atmospheric data in Common Data Format (CDF), and visualization of complex data with the National Space Science Data Center Graphics System. This would allow NASA to quickly explore the utility of such a system, given the ability to transfer data in and out of KGIS easily. The use and maintenance of the object hierarchies as polymorphic data types brings, to data management, a whole new set of problems and issues, few of which have been explored above the prototype level.

Author

**N89-26599*# National Aeronautics and Space Administration.
Goddard Space Flight Center, Greenbelt, MD.**

**THE UTILIZATION OF NEURAL NETS IN
POPULATING AN OBJECT-ORIENTED DATABASE**

WILLIAM J. CAMPBELL, SCOTT E. HILL, and ROBERT F. CROMP In its The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 249-263 Apr. 1989

Avail: CASI HC A03/MF A04

Existing NASA supported scientific data bases are usually developed, managed and populated in a tedious, error prone and self-limiting way in terms of what can be described in a relational Data Base Management System (DBMS). The next generation Earth remote sensing platforms (i.e., Earth Observation System, (EOS), will be capable of generating data at a rate of over 300 Mbs per second from a suite of instruments designed for different applications. What is needed is an innovative approach that creates object-oriented databases that segment, characterize, catalog and are manageable in a domain-specific context and whose contents are available interactively and in near-real-time to the user community. Described here is work in progress that utilizes an artificial neural net approach to characterize satellite imagery of undefined objects into high-level data objects. The characterized data is then dynamically allocated to an object-oriented data base where it can be reviewed and assessed by a user. The definition, development, and evolution of the overall data system model are steps in the creation of an application-driven knowledge-based scientific information system.

Author

N89-26600*# Martin Marietta Corp., Denver, CO. Information and Communications Systems.

A RAPID PROTOTYPING/ARTIFICIAL INTELLIGENCE APPROACH TO SPACE STATION-ERA INFORMATION MANAGEMENT AND ACCESS

RICHARD S. CARNAHAN, JR., STEPHEN M. COREY, and JOHN B. SNOW In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 265-279 Apr. 1989

Avail: CASI HC A03/MF A04

Applications of rapid prototyping and Artificial Intelligence techniques to problems associated with Space Station-era information management systems are described. In particular, the work is centered on issues related to: (1) intelligent man-machine interfaces applied to scientific data user support, and (2) the requirement that intelligent information management systems (IIMS) be able to efficiently process metadata updates concerning types of data handled. The advanced IIMS represents functional capabilities driven almost entirely by the needs of potential users. Space Station-era scientific data projected to be generated is likely to be significantly greater than data currently processed and analyzed. Information about scientific data must be presented clearly, concisely, and with support features to allow users at all levels of expertise efficient and cost-effective data access. Additionally, mechanisms for allowing more efficient IIMS metadata update processes must be addressed. The work reported covers the following IIMS design aspects: IIMS data and metadata modeling, including the automatic updating of IIMS-contained metadata, IIMS user-system interface considerations, including significant problems associated with remote access, user profiles, and on-line tutorial capabilities, and development of an IIMS query and browse facility, including the capability to deal with spatial information. A working prototype has been developed and is being enhanced.

Author

N89-26601*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

AN INTELLIGENT USER INTERFACE FOR BROWSING SATELLITE DATA CATALOGS

ROBERT F. CROMP and SHARON CROOK In its The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 281-299 Apr. 1989

Avail: CASI HC A03/MF A04

A large scale domain-independent spatial data management expert system that serves as a front-end to databases containing spatial data is described. This system is unique for two reasons. First, it uses spatial search techniques to generate a list of all the primary keys that fall within a user's spatial constraints prior to invoking the database management system, thus substantially decreasing the amount of time required to answer a user's query. Second, a domain-independent query expert system uses a domain-specific rule base to preprocess the user's English query, effectively mapping a broad class of queries into a smaller subset that can be handled by a commercial natural language processing system. The methods used by the spatial search module and the query expert system are explained, and the system architecture for the spatial data management expert system is described. The system is applied to data from the International Ultraviolet Explorer (IUE) satellite, and results are given.

Author

N89-26602*# Boeing Aerospace Co., Seattle, WA. Advanced Technology Center.

NATURAL LANGUAGE PROCESSING AND ADVANCED INFORMATION MANAGEMENT

JAMES E. HOARD In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 301-315 Apr. 1989

Avail: CASI HC A03/MF A04

Integrating diverse information sources and application software in a principled and general manner will require a very capable advanced information management (AIM) system. In particular, such a system will need a comprehensive addressing scheme to locate the material in its docuverse. It will also need a natural language processing (NLP) system of great sophistication. It seems that the NLP system must serve three functions. First, it provides an natural language interface (NLI) for the users. Second, it serves as the core component that understands and makes use of the real-world interpretations (RWIs) contained in the docuverse. Third, it enables the reasoning specialists (RSs) to arrive at conclusions that can be transformed into procedures that will satisfy the users' requests. The best candidate for an intelligent agent that can satisfactorily make use of RSs and transform documents (TDs) appears to be an object oriented data base (OODB). OODBs have, apparently, an inherent capacity to use the large numbers of RSs and TDs that will be required by an AIM system and an inherent capacity to use them in an effective way.

Author

N89-26603*# George Washington Univ., Washington, DC. Inst. for Artificial Intelligence.

INTEGRATION OF PERCEPTION AND REASONING IN FAST NEURAL MODULES

DAVID G. FRITZ In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 349-356 Apr. 1989 Prepared in cooperation with Cognitive Information Systems Co., Silver Spring, MD

Avail: CASI HC A02/MF A04

Artificial neural systems promise to integrate symbolic and sub-symbolic processing to achieve real time control of physical systems. Two potential alternatives exist. In one, neural nets can be used to front-end expert systems. The expert systems, in turn, are developed with varying degrees of parallelism, including their implementation in neural nets. In the other, rule-based reasoning and sensor data can be integrated within a single hybrid neural system. The hybrid system reacts as a unit to provide decisions (problem solutions) based on the simultaneous evaluation of data and rules. Discussed here is a model hybrid system based on the fuzzy cognitive map (FCM). The operation of the model is illustrated with the control of a hypothetical satellite that intelligently alters its attitude in space in response to an intersecting micrometeorite shower.

Author

N89-26604*# Maryland Univ., College Park, MD. Dept. of Computer Science.

A CONNECTIONIST MODEL FOR DYNAMIC CONTROL

KEVIN C. WHITFIELD, SHARON M. GOODALL, and JAMES A. REGGIA In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 357-370 Apr. 1989
(Contract NAG1-885; NSF IRI-84-51430)

Avail: CASI HC A03/MF A04

The application of a connectionist modeling method known as competition-based spreading activation to a camera tracking task is described. The potential is explored for automation of control and planning applications using

connectionist technology. The emphasis is on applications suitable for use in the NASA Space Station and in related space activities. The results are quite general and could be applicable to control systems in general.

Author

N89-26605*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

EXPERT SYSTEM DEVELOPMENT METHODOLOGY AND THE TRANSITION FROM PROTOTYPING TO OPERATIONS: FIESTA, A CASE STUDY

NADINE HAPPELL (Stanford Telecommunications, Inc., Reston, VA.), STEVE MIKSELL (Stanford Telecommunications, Inc., Reston, VA.), and CANDACE CARLISLE In its The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 373-383 Apr. 1989

Avail: CASI HC A03/MF A04

A major barrier in taking expert systems from prototype to operational status involves instilling end user confidence in the operational system. The software of different life cycle models is examined and the advantages and disadvantages of each when applied to expert system development are explored. The Fault Isolation Expert System for Tracking and data relay satellite system Applications (FIESTA) is presented as a case study of development of an expert system. The end user confidence necessary for operational use of this system is accentuated by the fact that it will handle real-time data in a secure environment, allowing little tolerance for errors. How FIESTA is dealing with transition problems as it moves from an off-line standalone prototype to an on-line real-time system is discussed.

Author

N89-26606*# National Aeronautics and Space Administration. Lyndon B. Johnson Space Center, Houston, TX.

APPLICATIONS OF FUZZY SETS TO RULE-BASED EXPERT SYSTEM DEVELOPMENT

ROBERT N. LEA In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 385-388 Apr. 1989

Avail: CASI HC A01/MF A04

Problems of implementing rule-based expert systems using fuzzy sets are considered. A fuzzy logic software development shell is used that allows inclusion of both crisp and fuzzy rules in decision making and process control problems. Results are given that compare this type of expert system to a human expert in some specific applications. Advantages and disadvantages of such systems are discussed.

Author

N89-26607*# Space Telescope Science Inst., Baltimore, MD.

GENETIC ALGORITHMS APPLIED TO THE SCHEDULING OF THE HUBBLE SPACE TELESCOPE

JEFFREY L. SPONSLER In NASA. Goddard Space Flight Center, The 1989 Goddard Conference on Space Applications of Artificial Intelligence p 391-402 Apr. 1989

Avail: CASI HC A03/MF A04

A prototype system employing a genetic algorithm (GA) has been developed to support the scheduling of the Hubble Space Telescope. A non-standard knowledge structure is used and appropriate genetic operators have been created. Several different crossover styles (random point selection, evolving points, and smart point selection) are tested and the best GA is compared with a neural network (NN) based optimizer.

The smart crossover operator produces the best results and the GA system is able to evolve complete schedules using it. The GA is not as time-efficient as the NN system and the NN solutions tend to be better.

Author

N90-22294*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE 1990 GODDARD CONFERENCE ON SPACE APPLICATIONS OF ARTIFICIAL INTELLIGENCE

JAMES L. RASH, ed. May 1990 342 p Conference held in Greenbelt, MD, 1-2 May 1990

(NASA-CP-3068; REPT-90B00078; NAS 1.55:3068)

Avail: CASI HC A15/MF A03

The papers presented at the 1990 Goddard Conference on Space Applications of Artificial Intelligence are given. The purpose of this annual conference is to provide a forum in which current research and development directed at space applications of artificial intelligence can be presented and discussed. The proceedings fall into the following areas: Planning and Scheduling, Fault Monitoring/Diagnosis, Image Processing and Machine Vision, Robotics/Intelligent Control, Development Methodologies, Information Management, and Knowledge Acquisition.

N90-22295*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE APPLICATION OF CONNECTIONISM TO QUERY PLANNING/SCHEDULING IN INTELLIGENT USER INTERFACES

NICHOLAS SHORT, JR. (National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.) and LOKENDRA SHASTRI (Pennsylvania Univ., Philadelphia.)

In its The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 3-16 May 1990

Avail: CASI HC A03/MF A03

In the mid nineties, the Earth Observing System (EOS) will generate an estimated 10 terabytes of data per day. This enormous amount of data will require the use of sophisticated technologies from real time distributed Artificial Intelligence (AI) and data management. Without regard to the overall problems in distributed AI, efficient models were developed for doing query planning and/or scheduling in intelligent user interfaces that reside in a network environment. Before intelligent query/planning can be done, a model for real time AI planning and/or scheduling must be developed. As Connectionist Models (CM) have shown promise in increasing run times, a connectionist approach to AI planning and/or scheduling is proposed. The solution involves merging a CM rule based system to a general spreading activation model for the generation and selection of plans. The system was implemented in the Rochester Connectionist Simulator and runs on a Sun 3/260.

Author

N90-22296*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

RESOURCE ALLOCATION PLANNING HELPER (RALPH): LESSONS LEARNED

RALPH DURHAM, NORMAN B. REILLY, and JOE B. SPRINGER In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 17-28 May 1990

Avail: CASI HC A03/MF A03

The current task of Resource Allocation Process includes the planning and apportionment of JPL's Ground Data System composed of the Deep Space Network and Mission

Control and Computing Center facilities. The addition of the data driven, rule based planning system, RALPH, has expanded the planning horizon from 8 weeks to 10 years and has resulted in large labor savings. Use of the system has also resulted in important improvements in science return through enhanced resource utilization. In addition, RALPH has been instrumental in supporting rapid turn around for an increased volume of special what if studies. The status of RALPH is briefly reviewed and important lessons learned from the creation of an highly functional design team are focused on through an evolutionary design and implementation period in which an AI shell was selected, prototyped, and ultimately abandoned, and through the fundamental changes to the very process that spawned the tool kit. Principal topics include proper integration of software tools within the planning environment, transition from prototype to delivered to delivered software, changes in the planning methodology as a result of evolving software capabilities and creation of the ability to develop and process generic requirements to allow planning flexibility. Author

N90-22297*# Martin Marietta Corp., Denver, CO. Information Systems Group.

THE ROLE OF ARTIFICIAL INTELLIGENCE TECHNIQUES IN SCHEDULING SYSTEMS

AMY L. GEOFFROY, DANIEL L. BRITT, and JOHN R. GOHRING In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 29-41 May 1990

Avail: CASI HC A03/MF A03

Artificial Intelligence (AI) techniques provide good solutions for many of the problems which are characteristic of scheduling applications. However, scheduling is a large, complex heterogeneous problem. Different applications will require different solutions. Any individual application will require the use of a variety of techniques, including both AI and conventional software methods. The operational context of the scheduling system will also play a large role in design considerations. The key is to identify those places where a specific AI technique is in fact the preferable solution, and to integrate that technique into the overall architecture. Author

N90-22298*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

PLAN-IT-2: THE NEXT GENERATION PLANNING AND SCHEDULING TOOL

WILLIAM C. EGGEMEYER and JENNIFER W. CRUZ In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 43-61 May 1990

Avail: CASI HC A03/MF A03

PLAN-IT is a scheduling program which has been demonstrated and evaluated in a variety of scheduling domains. The capability enhancements being made for the next generation of PLAN-IT, called PLAN-IT-2 is discussed. PLAN-IT-2 represents a complete rewrite of the original PLAN-IT incorporating major changes as suggested by the application experiences with the original PLAN-IT. A few of the enhancements described are additional types of constraints, such as states and resettable-depletables (batteries), dependencies between constraints, multiple levels of activity planning during the scheduling process, pattern constraint searching for opportunities as opposed to just minimizing the amount of conflicts, additional customization construction features for display and handling of diverse multiple time systems, and

reduction in both the size and the complexity for creating the knowledge-base to address the different problem domains.

Author

N90-22299*# Space Telescope Science Inst., Baltimore, MD. AN APPROACH TO RESCHEDULING ACTIVITIES BASED ON DETERMINATION OF PRIORITY AND DISRUPTIVITY

JEFFREY L. SPONSLER and MARK D. JOHNSTON In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 63-74 May 1990

Avail: CASI HC A03/MF A03

A constraint-based scheduling system called SPIKE is being used to create long term schedules for the Hubble Space Telescope. Feedback for the spacecraft or from other ground support systems may invalidate some scheduling decisions and those activities concerned must be reconsidered. A function rescheduling priority is defined which for a given activity performs a heuristic analysis and produces a relative numerical value which is used to rank all such entities in the order that they should be rescheduled. A function disruptivity is also defined that is used to place a relative numeric value on how much a pre-existing schedule would be changed in order to reschedule an activity. Using these functions, two algorithms (a stochastic neural network approach and an exhaustive search approach) are proposed to find the best place to reschedule an activity. Prototypes were implemented and preliminary testing reveals that the exhaustive technique produces only marginally better results at much greater computational cost. Author

N90-22300*# General Electric Co., Moorestown, NJ. Advanced Technology Labs.

SPACE COMMUNICATIONS SCHEDULER: A RULE-BASED APPROACH TO ADAPTIVE DEADLINE SCHEDULING

NICHOLAS STRAGUZZI In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 75-93 May 1990

Avail: CASI HC A03/MF A03

Job scheduling is a deceptively complex subfield of computer science. The highly combinatorial nature of the problem, which is NP-complete in nearly all cases, requires a scheduling program to intelligently transverse an immense search tree to create the best possible schedule in a minimal amount of time. In addition, the program must continually make adjustments to the initial schedule when faced with last-minute user requests, cancellations, unexpected device failures, quests, cancellations, unexpected device failures, etc. A good scheduler must be quick, flexible, and efficient, even at the expense of generating slightly less-than-optimal schedules. The Space Communication Scheduler (SCS) is an intelligent rule-based scheduling system. SCS is an adaptive deadline scheduler which allocates modular communications resources to meet an ordered set of user-specified job requests on board the NASA Space Station. SCS uses pattern matching techniques to detect potential conflicts through algorithmic and heuristic means. As a result, the system generates and maintains high density schedules without relying heavily on backtracking or blind search techniques. SCS is suitable for many common real-world applications. Author

N90-22301*# Rockwell International Corp., Houston, TX. Space Operations.

CONSTRAINT-BASED EVALUATION OF SEQUENTIAL PROCEDURES

MATTHEW R. BARRY In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 95-103 May 1990
Avail: CASI HC A02/MF A03

Constraining the operation of an agent requires knowledge of the restrictions to physical and temporal capabilities of that agent, as well as an inherent understanding of the desires being processed by that agent. Usually a set of constraints are available that must be adhered to in order to foster safe operations. In the worst case, violation of a constraint may be cause to terminate operation. If the agent is carrying out a plan, then a method for predicting the agent's desires, and therefore possible constraint violations, is required. The conceptualization of constraint-based reasoning used herein assumes that a system knows how to select a constraint for application as well as how to apply that constraint once it is selected. The application of constraint-based reasoning for evaluating certain kinds of plans known as sequential procedures is discussed. By decomposing these plans, it is possible to apply context dependent constraints in production system fashion without incorporating knowledge of the original planning process.

Author

N90-22302*# Colorado Univ., Boulder, CO. Lab. for Atmospheric and Space Physics.

SURE (SCIENCE USER RESOURCE EXPERT): A SCIENCE PLANNING AND SCHEDULING ASSISTANT FOR A RESOURCE BASED ENVIRONMENT

NANCY E. THALMAN and THOMAS P. SPARN In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 105-113 May 1990

Avail: CASI HC A02/MF A03

SURE (Science User Resource Expert) is one of three components that compose the SURPASS (Science User Resource Planning and Scheduling System). This system is a planning and scheduling tool which supports distributed planning and scheduling, based on resource allocation and optimization. Currently SURE is being used within the SURPASS by the UARS (Upper Atmospheric Research Satellite) SOLSTICE instrument to build a daily science plan and activity schedule and in a prototyping effort with NASA GSFC to demonstrate distributed planning and scheduling for the SOLSTICE II instrument on the EOS platform. For the SOLSTICE application the SURE utilizes a rule-based system. Development of a rule-based program using Ada CLIPS as opposed to using conventional programming, allows for capture of the science planning and scheduling heuristics in rules and provides flexibility in inserting or removing rules as the scientific objectives and mission constraints change. The SURE system's role as a component in the SURPASS, the purpose of the SURE planning and scheduling tool, the SURE knowledge base, and the software architecture of the SURE component are described.

Author

N90-22303*# Howard Univ., Washington, DC. Dept. of Electrical Engineering.

A KNOWLEDGE-BASED APPROACH TO IMPROVING OPTIMIZATION TECHNIQUES IN SYSTEM PLANNING

J. A. MOMOH and Z. Z. ZHANG In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 115-122 May 1990

Sponsored by DOE and Los Angeles Dept. of Water and Power, CA

(Contract NSF ECS-86-57559)

Avail: CASI HC A02/MF A03

A knowledge-based (KB) approach to improve mathematical programming techniques used in the system planning environment is presented. The KB system assists in selecting appropriate optimization algorithms, objective functions, constraints and parameters. The scheme is implemented by integrating symbolic computation of rules derived from operator and planner's experience and is used for generalized optimization packages. The KB optimization software package is capable of improving the overall planning process which includes correction of given violations. The method was demonstrated on a large scale power system discussed in the paper.

Author

N90-22304*# Martin Marietta Corp., Denver, CO. Information Systems Group.

MANAGING TEMPORAL RELATIONS

DANIEL L. BRITT, AMY L. GEOFFROY, and JOHN R. GOHRING In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 123-134 May 1990

Avail: CASI HC A03/MF A03

Various temporal constraints on the execution of activities are described, and their representation in the scheduling system MAESTRO is discussed. Initial examples are presented using a sample activity described. Those examples are expanded to include a second activity, and the types of temporal constraints that can obtain between two activities are explored. Soft constraints, or preferences, in activity placement are discussed. Multiple performances of activities are considered, with respect to both hard and soft constraints. The primary methods used in MAESTRO to handle temporal constraints are described as are certain aspects of contingency handling with respect to temporal constraints. A discussion of the overall approach, with indications of future directions for this research, concludes the study.

Author

N90-22305*# Ford Aerospace Corp., Seabrook, MD.

NASA GROUND TERMINAL COMMUNICATION EQUIPMENT AUTOMATED FAULT ISOLATION EXPERT SYSTEMS

Y. K. TANG and C. R. WETZEL In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 139-145 May 1990

Avail: CASI HC A02/MF A03

The prototype expert systems are described that diagnose the Distribution and Switching System I and II (DSS1 and DSS2), Statistical Multiplexers (SM), and Multiplexer and Demultiplexer systems (MDM) at the NASA Ground Terminal (NGT). A system level fault isolation expert system monitors the activities of a selected data stream, verifies that the fault exists in the NGT and identifies the faulty equipment. Equipment level fault isolation expert systems are invoked to isolate the fault to a Line Replaceable Unit (LRU) level. Input and sometimes output data stream activities for the equipment are available. The system level fault isolation expert system compares the equipment input and output status for a data stream and performs loopback tests (if necessary) to isolate the faulty equipment. The equipment level fault isolation system utilizes the process of elimination and/or the maintenance personnel's fault isolation experience stored in its knowledge base. The DSS1, DSS2 and

SM fault isolation systems, using the knowledge of the current equipment configuration and the equipment circuitry issues a set of test connections according to the predefined rules. The faulty component or board can be identified by the expert system by analyzing the test results. The MDM fault isolation system correlates the failure symptoms with the faulty component based on maintenance personnel experience. The faulty component can be determined by knowing the failure symptoms. The DSS1, DSS2, SM, and MDM equipment simulators are implemented in PASCAL. The DSS1 fault isolation expert system was converted to C language from VP-Expert and integrated into the NGT automation software for offline switch diagnoses. Potentially, the NGT fault isolation algorithms can be used for the DSS1, SM, and MDM located at Goddard Space Flight Center (GSFC).

Author

N90-22306*# National Aeronautics and Space Administration. Lewis Research Center, Cleveland, OH.

AUTONOMOUS POWER EXPERT SYSTEM

JERRY L. WALTERS (National Aeronautics and Space Administration, Lewis Research Center, Cleveland, OH.), EDWARD J. PETRIK (National Aeronautics and Space Administration, Lewis Research Center, Cleveland, OH.), MARY ELLEN ROTH (National Aeronautics and Space Administration, Lewis Research Center, Cleveland, OH.), LONG VAN TRUONG (National Aeronautics and Space Administration, Lewis Research Center, Cleveland, OH.), TODD QUINN (Sverdrup Technology, Inc., Cleveland, OH.), and WALTER M. KRAWCZONEK (Sverdrup Technology, Inc., Cleveland, OH.) In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 147-156 May 1990

Avail: CASI HC A02/MF A03

The Autonomous Power Expert (APEX) system was designed to monitor and diagnose fault conditions that occur within the Space Station Freedom Electrical Power System (SSF/EPS) Testbed. APEX is designed to interface with SSF/EPS testbed power management controllers to provide enhanced autonomous operation and control capability. The APEX architecture consists of three components: (1) a rule-based expert system, (2) a testbed data acquisition interface, and (3) a power scheduler interface. Fault detection, fault isolation, justification of probable causes, recommended actions, and incipient fault analysis are the main functions of the expert system component. The data acquisition component requests and receives pertinent parametric values from the EPS testbed and asserts the values into a knowledge base. Power load profile information is obtained from a remote scheduler through the power scheduler interface component. The current APEX design and development work is discussed. Operation and use of APEX by way of the user interface screens is also covered. Author

N90-22307*# TRW, Inc., Redondo Beach, CA. SPACECRAFT COMMAND VERIFICATION: THE AI SOLUTION

LORRAINE M. FESQ (TRW Defense and Space Systems Group, Redondo Beach, CA.), AMY STEPHAN (TRW Defense and Space Systems Group, Redondo Beach, CA.), and BRIAN K. SMITH (California Univ., Los Angeles.) In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 157-163 May 1990

Avail: CASI HC A02/MF A03

Recently, a knowledge-based approach was used to develop a system called the Command Constraint Checker (CCC)

for TRW. CCC was created to automate the process of verifying spacecraft command sequences. To check command files by hand for timing and sequencing errors is a time-consuming and error-prone task. Conventional software solutions were rejected when it was estimated that it would require 36 man-months to build an automated tool to check constraints by conventional methods. Using rule-based representation to model the various timing and sequencing constraints of the spacecraft, CCC was developed and tested in only three months. By applying artificial intelligence techniques, CCC designers were able to demonstrate the viability of AI as a tool to transform difficult problems into easily managed tasks. The design considerations used in developing CCC are discussed and the potential impact of this system on future satellite programs is examined.

Author

N90-22308*# Rockwell International Corp., Houston, TX. Space Operations.

ANALYZING SPACECRAFT CONFIGURATIONS THROUGH SPECIALIZATION AND DEFAULT REASONING

MATTHEW R. BARRY and CARLYLE M. LOWE In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 165-179 May 1990

Avail: CASI HC A03/MF A03

For an intelligent system to describe a real-world situation using as few statements as possible, it is necessary to make inferences based on observed data and to incorporate general knowledge of the reasoning domain into the description. These reasoning processes must reduce several levels of specific descriptions into only those few that most precisely describe the situation. Moreover, the system must be able to generate descriptions in the absence of data, as instructed by certain rules of inference. The deductions applied by the system, then, generate a high-level description from the low-level evidence provided by the real and default data sources. An implementation of these ideas in a real-world situation is described. The application concerns evaluation of Space Shuttle electro-mechanical system configurations by console operators in the Mission Control Center. A production system provides the reasoning mechanism through which the default assignments and specializations occur. Examples are provided within this domain for each type of inference, and the suitability is discussed of each toward achieving the goal of describing a situation in the fewest statements possible. Finally, several enhancements are suggested that will further increase the intelligence of similar spacecraft monitoring applications.

Author

N90-22309*# College of William and Mary, Williamsburg, VA. Dept. of Computer Science.

SIMULATION-BASED REASONING ABOUT THE PHYSICAL PROPAGATION OF FAULT EFFECTS

STEFAN FEYOCK and DALU LI In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 181-189 May 1990 (Contract NCC1-122)

Avail: CASI HC A02/MF A03

The research described deals with the effects of faults on complex physical systems, with particular emphasis on aircraft and spacecraft systems. Given that a malfunction has occurred and been diagnosed, the goal is to determine how that fault will propagate to other subsystems, and what the effects will

be on vehicle functionality. In particular, the use of qualitative spatial simulation to determine the physical propagation of fault effects in 3-D space is described. Author

N90-22310*# Howard Univ., Washington, DC. Dept. of Electrical Engineering.

KNOWLEDGE-BASED AND INTEGRATED MONITORING AND DIAGNOSIS IN AUTONOMOUS POWER SYSTEMS

J. A. MOMOH and Z. Z. ZHANG In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 191-199 May 1990 Spons. by DOE and Los Angeles Dept. of Water and Power, CA (Contract NSF ECS-86-57559)

Avail: CASI HC A02/MF A03

A new technique of knowledge-based and integrated monitoring and diagnosis (KBIMD) to deal with abnormalities and incipient or potential failures in autonomous power systems is presented. The KBIMD conception is discussed as a new function of autonomous power system automation. Available diagnostic modelling, system structure, principles and strategies are suggested. In order to verify the feasibility of the KBIMD, a preliminary prototype expert system is designed to simulate the KBIMD function in a main electric network of the autonomous power system. Author

N90-22311*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE PROCEDURE SAFETY SYSTEM

MAUREEN E. O'BRIEN In its The 1990 Goddard Conf. on Space Applications of Artificial Intelligence p 201-209 May 1990

Avail: CASI HC A02/MF A03

Telerobotic operations, whether under autonomous or teleoperated control, require a much more sophisticated safety system than that needed for most industrial applications. Industrial robots generally perform very repetitive tasks in a controlled, static environment. The safety system in that case can be as simple as shutting down the robot if a human enters the work area, or even simply building a cage around the work space. Telerobotic operations, however, will take place in a dynamic, sometimes unpredictable environment, and will involve complicated and perhaps unrehearsed manipulations. This creates a much greater potential for damage to the robot or objects in its vicinity. The Procedural Safety System (PSS) collects data from external sensors and the robot, then processes it through an expert system shell to determine whether an unsafe condition or potential unsafe condition exists. Unsafe conditions could include exceeding velocity, acceleration, torque, or joint limits, imminent collision, exceeding temperature limits, and robot or sensor component failure. If a threat to safety exists, the operator is warned. If the threat is serious enough, the robot is halted. The PSS, therefore, uses expert system technology to enhance safety thus reducing operator work load, allowing him/her to focus on performing the task at hand without the distraction of worrying about violating safety criteria. Author

N90-22312*# National Aeronautics and Space Administration. John F. Kennedy Space Center, Cocoa Beach, FL.

THE JPL/KSC TELEROBOTIC INSPECTION DEMONSTRATION

DAVID MITTMAN (Jet Propulsion Lab., California Inst. of Tech., Pasadena.), BRUCE BON (Jet Propulsion Lab., California Inst. of Tech., Pasadena.), CAROL COLLINS (Jet Propulsion

Lab., California Inst. of Tech., Pasadena.), GERRY FLEISCHER (Jet Propulsion Lab., California Inst. of Tech., Pasadena.), TODD LITWIN (Jet Propulsion Lab., California Inst. of Tech., Pasadena.), JACK MORRISON (Jet Propulsion Lab., California Inst. of Tech., Pasadena.), JACQUIE OMEARA (Jet Propulsion Lab., California Inst. of Tech., Pasadena.), STEPHEN PETERS (Jet Propulsion Lab., California Inst. of Tech., Pasadena.), JOHN BROGDON (National Aeronautics and Space Administration. John F. Kennedy Space Center, Cocoa Beach, FL.), BOB HUMENIUK et al. In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 213-222 May 1990

Avail: CASI HC A02/MF A03

An ASEA IRB90 robotic manipulator with attached inspection cameras was moved through a Space Shuttle Payload Assist Module (PAM) Cradle under computer control. The Operator and Operator Control Station, including graphics simulation, gross-motion spatial planning, and machine vision processing, were located at JPL. The Safety and Support personnel, PAM Cradle, IRB90, and image acquisition system, were stationed at the Kennedy Space Center (KSC). Images captured at KSC were used both for processing by a machine vision system at JPL, and for inspection by the JPL Operator. The system found collision-free paths through the PAM Cradle, demonstrated accurate knowledge of the location of both objects of interest and obstacles, and operated with a communication delay of two seconds. Safe operation of the IRB90 near Shuttle flight hardware was obtained both through the use of a gross-motion spatial planner developed at JPL using artificial intelligence techniques, and infrared beams and pressure sensitive strips mounted to the critical surfaces of the flight hardware at KSC. The Demonstration showed that telerobotics is effective for real tasks, safe for personnel and hardware, and highly productive and reliable for Shuttle payload operations and Space Station external operations. Author

N90-22313*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA. Sequence Automation Research Group.

SYSTEM CONTROL OF AN AUTONOMOUS PLANETARY MOBILE SPACECRAFT

WILLIAM C. DIAS and BARBARA A. ZIMMERMAN In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 223-237 May 1990

Avail: CASI HC A03/MF A03

The goal is to suggest the scheduling and control functions necessary for accomplishing mission objectives of a fairly autonomous interplanetary mobile spacecraft, while maximizing reliability. Goals are to provide an extensible, reliable system conservative in its use of on-board resources, while getting full value from subsystem autonomy, and avoiding the lure of ground micromanagement. A functional layout consisting of four basic elements is proposed: GROUND and SYSTEM EXECUTIVE system functions and RESOURCE CONTROL and ACTIVITY MANAGER subsystem functions. The system executive includes six subfunctions: SYSTEM MANAGER, SYSTEM FAULT PROTECTION, PLANNER, SCHEDULE ADAPTER, EVENT MONITOR and RESOURCE MONITOR. The full configuration is needed for autonomous operation on Moon or Mars, whereas a reduced version without the planning, schedule adaption and event monitoring functions could be appropriate for lower-autonomy use on the Moon. An implementation concept is suggested which is conservative in use of system resources and consists of modules combined with a

network communications fabric. A language concept termed a scheduling calculus for rapidly performing essential on-board schedule adaption functions is introduced. Author

N90-22314* TRW, Inc., Redondo Beach, CA.

AUTOMATED PROCEDURE EXECUTION FOR SPACE VEHICLE AUTONOMOUS CONTROL

THOMAS A. BRODEN and DAVID A. BROWN In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 239-250 May 1990

Avail: CASI HC A03/MF A03

Increased operational autonomy and reduced operating costs have become critical design objectives in next-generation NASA and DoD space programs. The objective is to develop a semi-automated system for intelligent spacecraft operations support. The Spacecraft Operations and Anomaly Resolution System (SOARS) is presented as a standardized, model-based architecture for performing High-Level Tasking, Status Monitoring and automated Procedure Execution Control for a variety of spacecraft. The particular focus is on the Procedure Execution Control module. A hierarchical procedure network is proposed as the fundamental means for specifying and representing arbitrary operational procedures. A separate procedure interpreter controls automatic execution of the procedure, taking into account the current status of the spacecraft as maintained in an object-oriented spacecraft model. Author

N90-22315* National Aeronautics and Space Administration, Goddard Space Flight Center, Greenbelt, MD.

USING EXPERT SYSTEMS TO IMPLEMENT A SEMANTIC DATA MODEL OF A LARGE MASS STORAGE SYSTEM

LARRY H. ROELOFS (Computer Technology Associates, Inc., Rockville, MD.) and WILLIAM J. CAMPBELL In its The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 253-270 May 1990

Avail: CASI HC A03/MF A03

The successful development of large volume data storage systems will depend not only on the ability of the designers to store data, but on the ability to manage such data once it is in the system. The hypothesis is that mass storage data management can only be implemented successfully based on highly intelligent meta data management services. There now exists a proposed mass store system standard proposed by the IEEE that addresses many of the issues related to the storage of large volumes of data, however, the model does not consider a major technical issue, namely the high level management of stored data. However, if the model were expanded to include the semantics and pragmatics of the data domain using a Semantic Data Model (SDM) concept, the result would be data that is expressive of the Intelligent Information Fusion (IIF) concept and also organized and classified in context to its use and purpose. The results are presented of a demonstration prototype SDM implemented using the expert system development tool NEXPERT OBJECT. In the prototype, a simple instance of a SDM was created to support a hypothetical application for the Earth Observing System, Data Information System (EOSDIS). The massive amounts of data that EOSDIS will manage requires the definition and design of a powerful information management system in order to support even the most basic needs of the project. The application domain is characterized by a semantic like network that represents the data content and the relationships between the data based on user views and the more generalized

domain architectural view of the information world. The data in the domain are represented by objects that define classes, types and instances of the data. In addition, data properties are selectively inherited between parent and daughter relationships in the domain. Based on the SDM a simple information system design is developed from the low level data storage media, through record management and meta data management to the user interface. Author

N90-22316* Martin Marietta Corp., Denver, CO. Information and Communications Systems.

KNOWLEDGE STRUCTURE REPRESENTATION AND AUTOMATED UPDATES IN INTELLIGENT INFORMATION MANAGEMENT SYSTEMS

STEPHEN COREY and RICHARD S. CARNAHAN, JR. In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 271-285 May 1990

Avail: CASI HC A03/MF A03

A continuing effort to apply rapid prototyping and Artificial Intelligence techniques to problems associated with projected Space Station-era information management systems is examined. In particular, timely updating of the various databases and knowledge structures within the proposed intelligent information management system (IIMS) is critical to support decision making processes. Because of the significantly large amounts of data entering the IIMS on a daily basis, information updates will need to be automatically performed with some systems requiring that data be incorporated and made available to users within a few hours. Meeting these demands depends first, on the design and implementation of information structures that are easily modified and expanded, and second, on the incorporation of intelligent automated update techniques that will allow meaningful information relationships to be established. Potential techniques are studied for developing such an automated update capability and IIMS update requirements are examined in light of results obtained from the IIMS prototyping effort. Author

N90-22317* Naval Research Lab., Washington, DC.

ADAPTIVE PATTERN RECOGNITION BY MINI-MAX NEURAL NETWORKS AS A PART OF AN INTELLIGENT PROCESSOR

HAROLD H. SZU In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 287-306 May 1990

Avail: CASI HC A03/MF A03

In this decade and progressing into 21st Century, NASA will have missions including Space Station and the Earth related Planet Sciences. To support these missions, a high degree of sophistication in machine automation and an increasing amount of data processing throughput rate are necessary. Meeting these challenges requires intelligent machines, designed to support the necessary automations in a remote space and hazardous environment. There are two approaches to designing these intelligent machines. One of these is the knowledge-based expert system approach, namely AI. The other is a non-rule approach based on parallel and distributed computing for adaptive fault-tolerances, namely Neural or Natural Intelligence (NI). The union of AI and NI is the solution to the problem stated above. The NI segment of this unit extracts features automatically by applying Cauchy simulated annealing to a mini-max cost energy function. The feature discovered by NI can then be passed to the AI system for future processing, and vice versa.

This passing increases reliability, for AI can follow the NI formulated algorithm exactly, and can provide the context knowledge base as the constraints of neurocomputing. The mini-max cost function that solves the unknown feature can furthermore give us a top-down architectural design of neural networks by means of Taylor series expansion of the cost function. A typical mini-max cost function consists of the sample variance of each class in the numerator, and separation of the center of each class in the denominator. Thus, when the total cost energy is minimized, the conflicting goals of intraclass clustering and interclass segregation are achieved simultaneously.

Author

N90-22318*# Texas Univ., Austin, TX. Dept. of Computer Sciences.

BOUNDED-TIME FAULT-TOLERANT RULE-BASED SYSTEMS

JAMES C. BROWNE, ALLEN EMERSON, MOHAMED GOUDA, DANIEL MIRANKER, ALOYSIUS MOK, and LOUIS ROSIER In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 307-321 May 1990 Sponsored in part by Texas Instruments, Inc.

(Contract N00014-89-J-1913; N00014-89-J-1472)

Avail: CASI HC A03/MF A03

Two systems concepts are introduced: bounded response-time and self-stabilization in the context of rule-based programs. These concepts are essential for the design of rule-based programs which must be highly fault tolerant and perform in a real time environment. The mechanical analysis of programs for these two properties is discussed. The techniques are used to analyze a NASA application.

Author

N90-22319*# Quebec Univ., Montreal (Quebec).

A KNOWLEDGE-BASED SYSTEM WITH LEARNING FOR COMPUTER COMMUNICATION NETWORK DESIGN

SAMUEL PIERRE (Quebec Univ., Montreal.), HAI HOC HOANG (Montreal Univ., Quebec), and EVELYNE TROPPER-HAUSEN In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 323-337 May 1990

Avail: CASI HC A03/MF A03

Computer communication network design is well-known as complex and hard. For that reason, the most effective methods used to solve it are heuristic. Weaknesses of these techniques are listed and a new approach based on artificial intelligence for solving this problem is presented. This approach is particularly recommended for large packet switched communication networks, in the sense that it permits a high degree of reliability and offers a very flexible environment dealing with many relevant design parameters such as link cost, link capacity, and message delay.

Author

N90-22320*# Computer Sciences Corp., Beltsville, MD.

EVALUATION OF A PROPOSED EXPERT SYSTEM DEVELOPMENT METHODOLOGY: TWO CASE STUDIES

LEWEY GILSTRAP In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 339-347 May 1990

Avail: CASI HC A02/MF A03

Two expert system development projects were studied to evaluate a proposed Expert Systems Development

Methodology (ESDM). The ESDM was developed to provide guidance to managers and technical personnel and serve as a standard in the development of expert systems. It was agreed that the proposed ESDM must be evaluated before it could be adopted; therefore a study was planned for its evaluation. This detailed study is now underway. Before the study began, however, two ongoing projects were selected for a retrospective evaluation. They were the Ranging Equipment Diagnostic Expert System (REDEX) and the Backup Control Mode Analysis and Utility System (BCAUS). Both projects were approximately 1 year into development. Interviews of project personnel were conducted, and the resulting data was used to prepare the retrospective evaluation. Decision models of the two projects were constructed and used to evaluate the completeness and accuracy of key provisions of ESDM. A major conclusion reached from these case studies is that suitability and risk analysis should be required for all AI projects, large and small. Further, the objectives of each stage of development during a project should be selected to reduce the next largest area of risk or uncertainty on the project.

Author

N90-22321*# Ford Aerospace Corp., Seabrook, MD. **SATELLITE IMAGE ANALYSIS USING NEURAL NETWORKS**

ROGER A. SHELDON In NASA, Goddard Space Flight Center, The 1990 Goddard Conference on Space Applications of Artificial Intelligence p 349-355 May 1990

Avail: CASI HC A02/MF A03

The tremendous backlog of unanalyzed satellite data necessitates the development of improved methods for data cataloging and analysis. Ford Aerospace has developed an image analysis system, SIANN (Satellite Image Analysis using Neural Networks) that integrates the technologies necessary to satisfy NASA's science data analysis requirements for the next generation of satellites. SIANN will enable scientists to train a neural network to recognize image data containing scenes of interest and then rapidly search data archives for all such images. The approach combines conventional image processing technology with recent advances in neural networks to provide improved classification capabilities. SIANN allows users to proceed through a four step process of image classification: filtering and enhancement, creation of neural network training data via application of feature extraction algorithms, configuring and training a neural network model, and classification of images by application of the trained neural network. A prototype experimentation testbed was completed and applied to climatological data.

Author

N90-70235* National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

PROCEEDINGS OF 1986 CONFERENCE ON ARTIFICIAL INTELLIGENCE APPLICATIONS

1986 144 p Conference held in Greenbelt, MD, 15 May 1986; sponsored by NASA, Goddard Space Flight Center, Greenbelt, MD and Bendix Field Engineering Corp., Greenbelt, MD (NASA-TM-101873; NAS 1.15:101873) Avail: CASI HC A07

The keynote address and ten technical papers presented at the 1986 Conference on Artificial Intelligence Applications are compiled. Most of the papers focused on various space science applications including ground control, mission operations support, data error detection, and command and control. Other papers addressed the development and testing of expert systems and the development of an intelligent user interface for large scientific databases.

N90-70236* National Aeronautics and Space Administration.
Goddard Space Flight Center, Greenbelt, MD.

THE DEVELOPMENT OF AN INTELLIGENT USER INTERFACE FOR NASA'S SCIENTIFIC DATABASES
WILLIAM J. CAMPBELL and LARRY H. ROELOFS
(Computer Technology Associates, Inc., McLean, VA.) In its Proceedings of 1986 Conference on Artificial Intelligence Applications p 8-19 1986

Avail: CASI HC A03

The National Space Science Data Center (NSSDC) has initiated an Intelligent Data Management (IDM) research effort which has as one of its components, the development of an Intelligent User Interface (IUI). The intent of the IUI effort is to develop a friendly and intelligent user interface service that is based on expert systems and natural language processing technologies. This paper presents the design concepts, development approach and evaluation of performance of a prototype Intelligent User Interface Subsystem (IUIS) supporting an operational database.

Author

N90-70237* Oak Ridge National Lab., TN.

AI GOES FORTH

W. B. DRESS In NASA, Goddard Space Flight Center, Proceedings of 1986 Conference on Artificial Intelligence Applications p 20-27 1986

(Contract DE-AC05-84-OR-21400)

Avail: CASI HC A02

The Forth language is presented as a vehicle for developing applications constrained by real-time considerations and size of hardware system. A specific example of rewriting OPS5 in a multitasking version of Forth shows that such applications can be extended to the realm of problems requiring a real-time artificial intelligence approach. The goal of high-speed, intelligent software operating in a restricted hardware environment is thus attainable in a cost-effective manner.

Author

N90-70238* IntelliTek, Inc., Silver Spring, MD.

THE EXPERT PROJECT MANAGEMENT SYSTEM (EPMS)

BARRY G. SILVERMAN and COTY DIAKITE In NASA, Goddard Space Flight Center, Proceedings of 1986 Conference on Artificial Intelligence Applications p 28-42 1986

Avail: CASI HC A03

Successful project managers (PMs) have been shown to rely on 'intuition,' experience, and analogical reasoning heuristics. For new PMs to be trained and experienced PMs to avoid repeating others' mistakes, it is necessary to make the knowledge and heuristics of successful PMs more widely available. The preparers have evolved a model of PM thought processes over the last decade that is now ready to be implemented as a generic PM aid. This aid consists of a series of 'specialist' expert systems (CRITIC, LIBRARIAN, IDEA MAN, CRAFTSMAN, and WRITER) that communicate with each other via a 'blackboard' architecture. The various specialist expert systems are driven to support PM training and problem solving since any 'answers' they pass to the blackboard are subjected to conflict identification (AGENDA FORMULATOR) and GOAL SETTER inference engines.

Author

N90-70239* Space Telescope Science Inst., Baltimore, MD.
AN EXPERT SYSTEM FOR GROUND SUPPORT OF THE HUBBLE SPACE TELESCOPE

DON ROSENTHAL, PATRICIA MONGER, GLENN MILLER

(Computer Sciences Corp., Baltimore, MD.), and MARK JOHNSTON In NASA, Goddard Space Flight Center, Proceedings of 1986 Conference on Artificial Intelligence Applications p 43-54 1986

Avail: CASI HC A03

The Hubble Space Telescope is an orbiting optical observatory due to be launched by the Space Shuttle in late 1987. It is a complex, multi-instrument observatory whose resources will be available to the world-wide astronomical community. The 'Transformation' system is a hybrid system which utilizes a rule-based expert system to convert scientific proposals into pre-optimized linked hierarchies of spacecraft activities. These activities are generated in a format that can be directly scheduled by the planning and scheduling component of the Space Telescope ground support system. The Transformation system will be described in detail in this paper, with particular attention given to the rule base.

Author

N90-70240* George Washington Univ., Washington, DC.
Dept. of Management Science.

AN EXPERT SYSTEM PROTOTYPE FOR AIDING IN THE DEVELOPMENT OF SOFTWARE FUNCTIONAL REQUIREMENTS FOR NASA GODDARD'S COMMAND MANAGEMENT SYSTEM: A CASE STUDY AND LESSONS LEARNED

JAY LIEBOWITZ In NASA, Goddard Space Flight Center, Proceedings of 1986 Conference on Artificial Intelligence Applications p 55-75 1986

(Contract NGT-09-010-800)

Avail: CASI HC A03

At NASA Goddard, the role of the command management system (CMS) is to transform general requests for spacecraft operations into detailed operational plans to be uplinked to the spacecraft. The CMS is part of the NASA Data System which entails the downlink of science and engineering data from NASA near-earth satellites to the user, and the uplink of command and control data to the spacecraft. Presently, it takes one to three years, with meetings once or twice a week, to determine functional requirements for CMS software design. As an alternative approach to the present technique of developing CMS software functional requirements, an expert system prototype was developed to aid in this function. Specifically, the knowledge base was formulated through interactions with domain experts, and was then linked to an existing expert system application generator called 'Knowledge Engineering System (Version 1.3).' Knowledge base development focused on four major steps: (1) develop the problem-oriented attribute hierarchy; (2) determine the knowledge management approach; (3) encode the knowledge base; and (4) validate, test, certify, and evaluate the knowledge base and the expert system prototype as a whole. Backcasting was accomplished for validating and testing the expert system prototype. Knowledge refinement, evaluation, and implementation procedures of the expert system prototype were then transacted.

Author

N90-70241* Bendix Field Engineering Corp., Lanham, MD.
THE DESIGN AND APPLICATION OF A TRANSPORTABLE INFERENCE ENGINE (TIE1)

DAVID R. MCLEAN In NASA, Goddard Space Flight Center, Proceedings of 1986 Conference on Artificial Intelligence Applications p 76-90 1986

(Contract NAS5-27772)

Avail: CASI HC A03

A Transportable Inference Engine (TIE1) system has

been developed by the author as part of the Interactive Experimenter Planning System (IEPS) task which is involved with developing expert systems in support of the Spacecraft Control Programs Branch at Goddard Space Flight Center in Greenbelt, Maryland. Unlike traditional inference engines, TIE1 is written in the C programming language. In the TIE1 system, knowledge is represented by a hierarchical network of objects which have rule frames. The TIE1 search algorithm uses a set of strategies, including backward chaining, to obtain the values of goals. The application of TIE1 to a spacecraft scheduling problem is described. This application involves the development of a strategies interpreter which uses TIE1 to do constraint checking.

Author

N90-70242* National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

A PROTOTYPE EXPERT SYSTEM IN OPS5 FOR DATA ERROR DETECTION

JAMES RASH In its Proceedings of 1986 Conference on Artificial Intelligence Applications p 91-105 1986

Avail: CASI HC A03

AA This prototype expert system, called Trajectory Preprocessing System (TRAPS), contains 49 rules and at present runs on an IBM PC in the OPS5+ software package from Artelligence, Inc. A prototype expert system has been developed in the OPS5 language to perform error checking on data which spacecraft builders/users supply to the NASA Goddard Space Flight Center for processing on the Communications Link Analysis and Simulation System (CLASS) computer. This prototype expert system, called Trajectory Preprocessing System (TRAPS), contains 49 rules and at present runs on an IBM PC in the OPS5+ software package from Artelligence, Inc. In its operational phase, TRAPS will run in the Oak Ridge Productional Language (ORPL) on the CLASS computer (a Perkin-Elmer 3244 supermini). ORPL, an implementation of OPS5 by the Oak Ridge National Laboratory in MULTIFORTH on a Hewlett-Packard 9836 desktop computer, is now being ported to SS-FORTH on the CLASS computer. This paper discusses the expert system problem domain, development approach, tools, results, and future plans stemming from the TRAPS project.

Author

N90-70243* Bendix Field Engineering Corp., Seabrook, MD. **MULTIPERSPECTIVE ANALYSIS AND TESTING OF EXPERT SYSTEMS**

TERRY B. BOLLINGER In NASA, Goddard Space Flight Center, Proceedings of 1986 Conference on Artificial Intelligence Applications p 106-113 1986

Avail: CASI HC A02

The paper describes a technique which the author developed for testing expert systems. The technique, which he calls multiperspective testing, can be applied during both the knowledge engineering phase and the acceptance testing phase of developing an expert system. The first step in multiperspective testing is to define a group of performance measures ('perspectives') that focus on the behavior of the knowledge base. For each such measure, the results of testing are summarized in four scores, which the author calls 'expansion,' 'detection,' 'discrimination,' and 'comprehension.' These scores have the advantage of providing more specific information about how the knowledge base should be updated or corrected.

Author

N90-70244* Maryland Univ., College Park, MD. Dept. of Computer Science.

SIMPLE METHODS OF EXPLOITING THE UNDERLYING STRUCTURE OF RULE-BASED SYSTEMS

JAMES HENDLER In NASA, Goddard Space Flight Center, Proceedings of 1986 Conference on Artificial Intelligence Applications p 116-121 1986

Avail: CASI HC A02

Much recent work in the field of expert systems research has aimed at exploiting the underlying structures of the rule base for reasons of analysis. Such techniques as Petri-nets and GAGs have been proposed as representational structures that will allow complete analysis. Much has been made of proving isomorphisms between the rule bases and the mechanisms, and in examining the theoretical power of this analysis. In this paper we describe some early work in a new system which has much simpler (and thus, one hopes, more easily achieved) aims and less formality. The technique being examined is a very simple one: OPS5 programs are analyzed in a purely syntactic way and a FSA description is generated. In this paper we describe the technique and some user interface tools which exploit this structure.

Derived from text

N90-70245* Computer Technology Associates, Inc., Lanham, MD.

DIOGENES: EXPERT SYSTEM FOR EXTRACTION OF DATA SYSTEM REQUIREMENTS

ROBERT W. HOBBS and T. PATRICK GORMAN In NASA, Goddard Space Flight Center, Proceedings of 1986 Conference on Artificial Intelligence Applications p 122-140 1986

Avail: CASI HC A03

The initial operations concept expresses information about system objectives, and defines the system users, system interfaces, and operational performance constraints. We have developed a prototype expert system which has established the feasibility of automating a scenario-driven methodology for deriving top-level specifications and preliminary designs for user data systems. This scenario-driven methodology uses an initial design, an initial operations concept, and user scenarios as the starting point for system definition. The top-level initial design is a functional description of the system in the form of an annotated data flow diagram. The initial operations concept expresses information about system objectives, and defines the system users, system interfaces, and operational performance constraints. The user scenarios are detailed time-lined descriptions of user activities, developed by prospective end users. These scenarios, along with the initial design and operations concept, are analyzed and iterated by the expert system to form a consistent set. The resulting User Scenario-Operation Set plays a key role in the development of requirements and system tests.

Derived from text

N91-22769# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE 1991 GODDARD CONFERENCE ON SPACE APPLICATIONS OF ARTIFICIAL INTELLIGENCE

JAMES L. RASH, ed. Washington May 1991 361 p Conference held in Greenbelt, MD, 13-15 May 1991 (NASA-CP-3110; REPT-91B00064; NAS 1.55:3110) Avail: CASI HC A16/MF A03

The purpose of this annual conference is to provide a forum in which current research and development directed at space applications of artificial intelligence can be presented and

discussed. The papers in this proceeding fall into the following areas: Planning and scheduling, fault monitoring/diagnosis/recovery, machine vision, robotics, system development, information management, knowledge acquisition and representation, distributed systems, tools, neural networks, and miscellaneous applications.

N91-22770*# Computer Resources International A/S (Denmark).

OPTIMUM-AIV: A PLANNING AND SCHEDULING SYSTEM FOR SPACECRAFT AIV

M. M. ARENTOFT (Computer Resources International A/S (Denmark).), JENS J. FUCHS (Computer Resources International A/S (Denmark).), Y. PARROD (MATRA Espace, Buc (France).), ANDRE GASQUET (MATRA Espace, Buc (France).), J. STADER (Edinburgh Univ. (Scotland).), I. STOKES (Progespace (France).), and H. VADON (European Space Agency. European Space Research and Technology Center, ESTEC, Noordwijk, Netherlands) In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 3-13 May 1991

Avail: CASI HC A03/MF A03

A project undertaken for the European Space Agency (ESA) is presented. The project is developing a knowledge based software system for planning and scheduling of activities for spacecraft assembly, integration, and verification (AIV). The system extends into the monitoring of plan execution and the plan repair phase. The objectives are to develop an operational kernel of a planning, scheduling, and plan repair tool, called OPTIMUM-AIV, and to provide facilities which will allow individual projects to customize the kernel to suit its specific needs. The kernel shall consist of a set of software functionalities for assistance in initial specification of the AIV plan, in verification and generation of valid plans and schedules for the AIV activities, and in interactive monitoring and execution problem recovery for the detailed AIV plans. Embedded in OPTIMUM-AIV are external interfaces which allow integration with alternative scheduling systems and project databases. The current status of the OPTIMUM-AIV project, as of Jan. 1991, is that a further analysis of the AIV domain has taken place through interviews with satellite AIV experts, a software requirement document (SRD) for the full operational tool was approved, and an architectural design document (ADD) for the kernel excluding external interfaces is ready for review.

Author

N91-22771*# Intelligent Systems Lab., Chantilly, VA.

TDRSS MOMENTUM UNLOAD PLANNING

GEORGE R. CROSS, MITCHELL A. POTTER, J. DOUGLASS WHITEHEAD, and JAMES T. SMITH In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 15-29 May 1991

Avail: CASI HC A03/MF A03

A knowledge-based system is described which monitors TDRSS telemetry for problems in the momentum unload procedure. The system displays TDRSS telemetry and commands in real time via X-windows. The system constructs a momentum unload plan which agrees with the preferences of the attitude control specialists and the momentum growth characteristics of the individual spacecraft. During the execution of the plan, the system monitors the progress of the procedure and watches for unexpected problems.

Author

N91-22772*# Martin Marietta Corp., Denver, CO. Information Systems Group.

SHARING INTELLIGENCE: DECISION-MAKING INTERACTIONS BETWEEN USERS AND SOFTWARE IN MAESTRO

AMY L. GEOFFROY, JOHN R. GOHRING, and DANIEL L. BRITT In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 31-44 May 1991

Avail: CASI HC A03/MF A03

By combining the best of automated and human decision-making in scheduling many advantages can accrue. The joint performance of the user and system is potentially much better than either alone. Features of the MAESTRO scheduling system serve to illustrate concepts of user/software cooperation. MAESTRO may be operated at a user-determinable and dynamic level of autonomy. Because the system allows so much flexibility in the allocation of decision-making responsibilities, and provides users with a wealth of information and other support for their own decision-making, better overall schedules may result.

Author

N91-22773*# Space Telescope Science Inst., Baltimore, MD.

TRANSFORMATION REBORN: A NEW GENERATION EXPERT SYSTEM FOR PLANNING HST OPERATIONS

ANDREW GERB In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 45-58 May 1991

Avail: CASI HC A03/MF A03

The Transformation expert system (TRANS) converts proposals for astronomical observations with the Hubble Space Telescope (HST) into detailed observing plans. It encodes expert knowledge to solve problems faced in planning and commanding HST observations to enable their processing by the Science Operations Ground System (SOGS). Among these problems are determining an acceptable order of executing observations, grouping of observations to enhance efficiency and schedulability, inserting extra observations when necessary, and providing parameters for commanding HST instruments. TRANS is currently an operational system and plays a critical role in the HST ground system. It was originally designed using forward-chaining provided by the OPS5 expert system language, but has been reimplemented using a procedural knowledge base. This reimplementation was forced by the explosion in the amount of OPS5 code required to specify the increasingly complicated situations requiring expert-level intervention by the TRANS knowledge base. This problem was compounded by the difficulty of avoiding unintended interaction between rules. To support the TRANS knowledge base, XCL, a small but powerful extension to Common Lisp was implemented. XCL allows a compact syntax for specifying assignments and references to object attributes. XCL also allows the capability to iterate over objects and perform keyed lookup. The reimplementation of TRANS has greatly diminished the effort needed to maintain and enhance it. As a result of this, its functions have been expanded to include warnings about observations that are difficult or impossible to schedule or command, providing data to aid SPIKE, an intelligent planning system used for HST long-term scheduling, and providing information to the Guide Star Selection System (GSSS) to aid in determination of the long range availability of guide stars.

Author

N91-22774*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

USING C TO BUILD A SATELLITE SCHEDULING EXPERT SYSTEM: EXAMPLES FROM THE

EXPLORER PLATFORM PLANNING SYSTEM

DAVID R. MCLEAN (Bendix Field Engineering Corp., Greenbelt, MD.), ALAN TUCHMAN (Bendix Field Engineering Corp., Greenbelt, MD.), and WILLIAM J. POTTER In its The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 59-69 May 1991
(Contract NAS5-31000; NAS5-27772)
Avail: CASI HC A03/MF A03

Recently, many expert systems were developed in a LISP environment and then ported to the real world C environment before the final system is delivered. This situation may require that the entire system be completely rewritten in C and may actually result in a system which is put together as quickly as possible with little regard for maintainability and further evolution. With the introduction of high performance UNIX and X-windows based workstations, a great deal of the advantages of developing a first system in the LISP environment have become questionable. A C-based AI development effort is described which is based on a software tools approach with emphasis on reusability and maintainability of code. The discussion starts with simple examples of how list processing can easily be implemented in C and then proceeds to the implementations of frames and objects which use dynamic memory allocation. The implementation of procedures which use depth first search, constraint propagation, context switching and a blackboard-like simulation environment are described. Techniques for managing the complexity of C-based AI software are noted, especially the object-oriented techniques of data encapsulation and incremental development. Finally, all these concepts are put together by describing the components of planning software called the Planning And Resource Reasoning (PARR) shell. This shell was successfully utilized for scheduling services of the Tracking and Data Relay Satellite System for the Earth Radiation Budget Satellite since May 1987 and will be used for operations scheduling of the Explorer Platform in November 1991.

Author

N91-22775*# LONG RANGE SCIENCE SCHEDULING FOR THE HUBBLE SPACE TELESCOPE

GLENN MILLER and MARK JOHNSTON In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 71-82 May 1991
Avail: CASI HC A03/MF A03

Observations with NASA's Hubble Space Telescope (HST) are scheduled with the assistance of a long-range scheduling system (SPIKE) that was developed using artificial intelligence techniques. In earlier papers, the system architecture and the constraint representation and propagation mechanisms were described. The development of high-level automated scheduling tools, including tools based on constraint satisfaction techniques and neural networks is described. The performance of these tools in scheduling HST observations is discussed.

Author

N91-22776*# AI TECHNIQUES FOR A SPACE APPLICATION SCHEDULING PROBLEM

N. THALMAN, T. SPARN, L. JAFFRES, D. GABLEHOUSE, D. JUDD, and C. RUSSELL In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 83-93 May 1991
Avail: CASI HC A02/MF A03

Scheduling is a very complex optimization problem

which can be categorized as an NP-complete problem. NP-complete problems are quite diverse, as are the algorithms used in searching for an optimal solution. In most cases, the best solutions that can be derived for these combinatorial explosive problems are near-optimal solutions. Due to the complexity of the scheduling problem, artificial intelligence (AI) can aid in solving these types of problems. Some of the factors are examined which make space application scheduling problems difficult and presents a fairly new AI-based technique called tabu search as applied to a real scheduling application. the specific problem is concerned with scheduling application. The specific problem is concerned with scheduling solar and stellar observations for the SOLar-STellar Irradiance Comparison Experiment (SOLSTICE) instrument in a constrained environment which produces minimum impact on the other instruments and maximizes target observation times. The SOLSTICE instrument will fly on-board the Upper Atmosphere Research Satellite (UARS) in 1991, and a similar instrument will fly on the earth observing system (Eos).

Author

N91-22777*# Mitre Corp., Houston, TX.

A FAILURE DIAGNOSIS AND IMPACT ASSESSMENT PROTOTYPE FOR SPACE STATION FREEDOM

CAROLYN G. BAKER and CHRISTOPHER A. MARSH In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 97-111 May 1991
(Contract NAS9-18057)

Avail: CASI HC A03/MF A03

NASA is investigating the use of advanced automation to enhance crew productivity for Space Station Freedom in numerous areas, one being failure management. A prototype is described that diagnoses failure sources and assesses the future impacts of those failures on other Freedom entities.

Author

N91-22778*# Mitre Corp., Houston, TX.

A FAILURE RECOVERY PLANNING PROTOTYPE FOR SPACE STATION FREEDOM

DAVID G. HAMMEN and CHRISTINE M. KELLY In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 113-127 May 1991
(Contract NAS9-18057)

Avail: CASI HC A03/MF A03

NASA is investigating the use of advanced automation to enhance crew productivity for Space Station Freedom in numerous areas, including failure management. A prototype is described that uses various advanced automation techniques to generate courses of action whose intents are to recover from a diagnosed failure, and to do so within the constraints levied by the failure and by Freedom's configuration and operating conditions.

Author

N91-22779*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE GENERIC SPACECRAFT ANALYST ASSISTANT (GENSAA): A TOOL FOR AUTOMATING SPACECRAFT MONITORING WITH EXPERT SYSTEMS

PETER M. HUGHES (National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.) and EDWARD C. LUCZAK (Computer Sciences Corp., Beltsville, MD.) In its The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 129-139 May 1991
Avail: CASI HC A03/MF A03

Flight Operations Analysts (FOAs) in the Payload Operations Control Center (POCC) are responsible for monitoring a satellite's health and safety. As satellites become more complex and data rates increase, FOAs are quickly approaching a level of information saturation. The FOAs in the spacecraft control center for the COBE (Cosmic Background Explorer) satellite are currently using a fault isolation expert system named the Communications Link Expert Assistance Resource (CLEAR), to assist in isolating and correcting communications link faults. Due to the success of CLEAR and several other systems in the control center domain, many other monitoring and fault isolation expert systems will likely be developed to support control center operations during the early 1990s. To facilitate the development of these systems, a project was initiated to develop a domain specific tool, named the Generic Spacecraft Analyst Assistant (GenSAA). GenSAA will enable spacecraft analysts to easily build simple real-time expert systems that perform spacecraft monitoring and fault isolation functions. Lessons learned during the development of several expert systems at Goddard, thereby establishing the foundation of GenSAA's objectives and offering insights in how problems may be avoided in future project, are described. This is followed by a description of the capabilities, architecture, and usage of GenSAA along with a discussion of its application to future NASA missions.

Author

N91-22780*# National Aeronautics and Space Administration. Lyndon B. Johnson Space Center, Houston, TX.

REPRESENTING FUNCTIONS/PROCEDURES AND PROCESSES/STRUCTURES FOR ANALYSIS OF EFFECTS OF FAILURES ON FUNCTIONS AND OPERATIONS

JANE T. MALIN (National Aeronautics and Space Administration, Lyndon B. Johnson Space Center, Houston, TX.) and DANIEL B. LEIFKER (Mitre Corp., Houston, TX.) In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 141-151 May 1991

Avail: CASI HC A03/MF A03

Current qualitative device and process models represent only the structure and behavior of physical systems. However, systems in the real world include goal-oriented activities that generally cannot be easily represented using current modeling techniques. An extension of a qualitative modeling system, known as functional modeling, which captures goal-oriented activities explicitly is proposed and how they may be used to support intelligent automation and fault management is shown.

Author

N91-22781*# Sverdrup Technology, Inc., Brook Park, OH. **AUTONOMOUS POWER SYSTEM INTELLIGENT DIAGNOSIS AND CONTROL**

MARK J. RINGER, TODD M. QUINN, and ANTHONY MEROLLA In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 153-167 May 1991

Avail: CASI HC A03/MF A03

The Autonomous Power System (APS) project at NASA Lewis Research Center is designed to demonstrate the abilities of integrated intelligent diagnosis, control, and scheduling techniques to space power distribution hardware. Knowledge-based software provides a robust method of control for highly complex space-based power systems that conventional methods do not allow. The project consists of three elements: the

Autonomous Power Expert System (APEX) for fault diagnosis and control, the Autonomous Intelligent Power Scheduler (AIPS) to determine system configuration, and power hardware (Brassboard) to simulate a space based power system. The operation of the Autonomous Power System as a whole is described and the responsibilities of the three elements - APEX, AIPS, and Brassboard - are characterized. A discussion of the methodologies used in each element is provided. Future plans are discussed for the growth of the Autonomous Power System.

Author

N91-22782*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

A MACHINE INDEPENDENT EXPERT SYSTEM FOR DIAGNOSING ENVIRONMENTALLY INDUCED SPACECRAFT ANOMALIES

MARK J. ROLINCIK In its The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 169-179 May 1991

Avail: CASI HC A03/MF A03

A new rule-based, machine independent analytical tool for diagnosing spacecraft anomalies, the EnviroNET expert system, was developed. Expert systems provide an effective method for storing knowledge, allow computers to sift through large amounts of data pinpointing significant parts, and most importantly, use heuristics in addition to algorithms which allow approximate reasoning and inference, and the ability to attack problems not rigidly defined. The EnviroNET expert system knowledge base currently contains over two hundred rules, and links to databases which include past environmental data, satellite data, and previous known anomalies. The environmental causes considered are bulk charging, single event upsets (SEU), surface charging, and total radiation dose.

Author

N91-22783*# Symbiotics, Inc., Cambridge, MA.

A HIERARCHICAL DISTRIBUTED CONTROL MODEL FOR COORDINATING INTELLIGENT SYSTEMS

RICHARD M. ADLER In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 183-197 May 1991

(Contract NAS10-11606; DAAB10-87-C-0053)

Avail: CASI HC A03/MF A03

A hierarchical distributed control (HDC) model for coordinating cooperative problem-solving among intelligent systems is described. The model was implemented using SOCIAL, an innovative object-oriented tool for integrating heterogeneous, distributed software systems. SOCIAL embeds applications in 'wrapper' objects called Agents, which supply predefined capabilities for distributed communication, control, data specification, and translation. The HDC model is realized in SOCIAL as a 'Manager'Agent that coordinates interactions among application Agents. The HDC Manager: indexes the capabilities of application Agents; routes request messages to suitable server Agents; and stores results in a commonly accessible 'Bulletin-Board'. This centralized control model is illustrated in a fault diagnosis application for launch operations support of the Space Shuttle fleet at NASA, Kennedy Space Center.

Author

N91-22784*# Martin Marietta Aerospace, Denver, CO.

MACHINE VISION BASED TELEOPERATION AID

WILLIAM A. HOFF, LANCE B. GATRELL, and JOHN R. SPOFFORD In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial

Intelligence p 199-213 May 1991

Avail: CASI HC A03/MF A03

When teleoperating a robot using video from a remote camera, it is difficult for the operator to gauge depth and orientation from a single view. In addition, there are situations where a camera mounted for viewing by the teleoperator during a teleoperation task may not be able to see the tool tip, or the viewing angle may not be intuitive (requiring extensive training to reduce the risk of incorrect or dangerous moves by the teleoperator). A machine vision based teleoperator aid is presented which uses the operator's camera view to compute an object's pose (position and orientation), and then overlays onto the operator's screen information on the object's current and desired positions. The operator can choose to display orientation and translation information as graphics and/or text. This aid provides easily assimilated depth and relative orientation information to the teleoperator. The camera may be mounted at any known orientation relative to the tool tip. A preliminary experiment with human operators was conducted and showed that task accuracies were significantly greater with than without this aid.

Author

N91-22785*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

A COLOR-CODED VISION SCHEME FOR ROBOTICS
KELLEY TINA JOHNSON In its The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 215-226 May 1991
(Contract NAG5-1382)

Avail: CASI HC A03/MF A03

Most vision systems for robotic applications rely entirely on the extraction of information from gray-level images. Humans, however, regularly depend on color to discriminate between objects. Therefore, the inclusion of color in a robot vision system seems a natural extension of the existing gray-level capabilities. A method for robot object recognition using a color-coding classification scheme is discussed. The scheme is based on an algebraic system in which a two-dimensional color image is represented as a polynomial of two variables. The system is then used to find the color contour of objects. In a controlled environment, such as that of the in-orbit space station, a particular class of objects can thus be quickly recognized by its color.

Author

N91-22786*# Computer Resources International A/S (Denmark).

EXPERT OPERATOR'S ASSOCIATE: A KNOWLEDGE BASED SYSTEM FOR SPACECRAFT CONTROL

MOGENS NIELSEN (Computer Resources International A/S (Denmark).), KLAUS GRUE (Computer Resources International A/S (Denmark).), and FRANCOIS LECOUAT (MATRA Espace, Toulouse, France) In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 227-237 May 1991
(Contract ESA-7627/88/NL/DG)

Avail: CASI HC A03/MF A03

The Expert Operator's Associate (EOA) project is presented which studies the applicability of expert systems for day-to-day space operations. A prototype expert system is developed, which operates on-line with an existing spacecraft control system at the European Space Operations Centre, and functions as an 'operator's assistant' in controlling satellites. The prototype is demonstrated using an existing real-time simulation model of the MARECS-B2 telecommunication satellite. By

developing a prototype system, the extent to which reliability and effectiveness of operations can be enhanced by AI based support is examined. In addition the study examines the questions of acquisition and representation of the 'knowledge' for such systems, and the feasibility of 'migration' of some (currently) ground-based functions into future spaceborne autonomous systems.

Author

N91-22787*# Computer Sciences Corp., Beltsville, MD.
VALIDATION AND VERIFICATION OF EXPERT SYSTEMS

LEWEY GILSTRAP In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 241-248 May 1991

Avail: CASI HC A02/MF A03

Validation and verification (V&V) are procedures used to evaluate system structure or behavior with respect to a set of requirements. Although expert systems are often developed as a series of prototypes without requirements, it is not possible to perform V&V on any system for which requirements have not been prepared. In addition, there are special problems associated with the evaluation of expert systems that do not arise in the evaluation of conventional systems, such as verification of the completeness and accuracy of the knowledge base. The criticality of most NASA missions make it important to be able to certify the performance of the expert systems used to support these mission. Recommendations for the most appropriate method for integrating V&V into the Expert System Development Methodology (ESDM) and suggestions for the most suitable approaches for each stage of ESDM development are presented.

Author

N91-22788*# National Aeronautics and Space Administration. Lewis Research Center, Cleveland, OH.

TECHNIQUES AND IMPLEMENTATION OF THE EMBEDDED RULE-BASED EXPERT SYSTEM USING ADA

EUGENE M. LIBERMAN (Sverdrup Technology, Inc., Brook Park, OH.) and ROBERT E. JONES In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 249-255 May 1991

Avail: CASI HC A02/MF A03

Ada is becoming an increasingly popular programming language for large Government-funded software projects. Ada with its portability, transportability, and maintainability lends itself well to today's complex programming environment. In addition, expert systems have also assured a growing role in providing human-like reasoning capability and expertise for computer systems. The integration of expert system technology with Ada programming language, specifically a rule-based expert system using an ART-Ada (Automated Reasoning Tool for Ada) system shell is discussed. The NASA Lewis Research Center was chosen as a beta test site for ART-Ada. The test was conducted by implementing the existing Autonomous Power EXpert System (APEX), a Lisp-base power expert system, in ART-Ada. Three components, the rule-based expert system, a graphics user interface, and communications software make up SMART-Ada (Systems fault Management with ART-Ada). The main objective, to conduct a beta test on the ART-Ada rule-based expert system shell, was achieved. The system is operational. New Ada tools will assist in future successful projects. ART-Ada is one such tool and is a viable alternative to the straight Ada code when an application requires a rule-based or knowledge-based approach.

B.G.

N91-22789*# Syscon Corp., Washington, DC.

**A REUSABLE KNOWLEDGE ACQUISITION SHELL:
KASH**

CHRISTOPHER WESTPHAL, STEPHEN WILLIAMS, and VIRGINIA KEECH In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 257-272 May 1991

Avail: CASI HC A03/MF A03

KASH (Knowledge Acquisition SHell) is proposed to assist a knowledge engineer by providing a set of utilities for constructing knowledge acquisition sessions based on interviewing techniques. The information elicited from domain experts during the sessions is guided by a question dependency graph (QDG). The QDG defined by the knowledge engineer, consists of a series of control questions about the domain that are used to organize the knowledge of an expert. The content information supplies by the expert, in response to the questions, is represented in the form of a concept map. These maps can be constructed in a top-down or bottom-up manner by the QDG and used by KASH to generate the rules for a large class of expert system domains. Additionally, the concept maps can support the representation of temporal knowledge. The high degree of reusability encountered in the QDG and concept maps can vastly reduce the development times and costs associated with producing intelligent decision aids, training programs, and process control functions.

Author

N91-22790*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

**CAPTURING FLIGHT SYSTEM TEST ENGINEERING
EXPERTISE: LESSONS LEARNED**

IRENE WONG WOERNER In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 273-283 May 1991

Avail: CASI HC A03/MF A03

Within a few years, JPL will be challenged by the most active mission set in history. Concurrently, flight systems are increasingly more complex. Presently, the knowledge to conduct integration and test of spacecraft and large instruments is held by a few key people, each with many years of experience. JPL is in danger of losing a significant amount of this critical expertise, through retirement, during a period when demand for this expertise is rapidly increasing. The most critical issue at hand is to collect and retain this expertise and develop tools that would ensure the ability to successfully perform the integration and test of future spacecraft and large instruments. The proposed solution was to capture and codify a subset of existing knowledge, and to utilize this captured expertise in knowledge-based systems. First year results and activities planned for the second year of this ongoing effort are described. Topics discussed include lessons learned in knowledge acquisition and elicitation techniques, life-cycle paradigms, and rapid prototyping of a knowledge-based advisor (Spacecraft Test Assistant) and a hypermedia browser (Test Engineering Browser). The prototype Spacecraft Test Assistant supports a subset of integration and test activities for flight systems. Browser is a hypermedia tool that allows users easy perusal of spacecraft test topics. A knowledge acquisition tool called ConceptFinder which was developed to search through large volumes of data for related concepts is also described and is modified to semi-automate the process of creating hypertext links.

Author

N91-22791*# Idaho National Eng. Lab., Idaho Falls, ID.

**A PROVEN KNOWLEDGE-BASED APPROACH TO
PRIORITIZING PROCESS INFORMATION**

DANIEL R. CORSBERG In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 287-293 May 1991

(Contract DE-AC07-76ID-01570)

Avail: CASI HC A02/MF A03

Many space-related processes are highly complex systems subject to sudden, major transients. In any complex process control system, a critical aspect is rapid analysis of the changing process information. During a disturbance, this task can overwhelm humans as well as computers. Humans deal with this by applying heuristics in determining significant information. A simple, knowledge-based approach to prioritizing information is described. The approach models those heuristics that humans would use in similar circumstances. The approach described has received two patents and was implemented in the Alarm Filtering System (AFS) at the Idaho National Engineering Laboratory (INEL). AFS was first developed for application in a nuclear reactor control room. It has since been used in chemical processing applications, where it has had a significant impact on control room environments. The approach uses knowledge-based heuristics to analyze data from process instrumentation and respond to that data according to knowledge encapsulated in objects and rules. While AFS cannot perform the complete diagnosis and control task, it has proven to be extremely effective at filtering and prioritizing information. AFS was used for over two years as a first level of analysis for human diagnosticians. Given the approach's proven track record in a wide variety of practical applications, it should be useful in both ground- and space-based systems.

Author

N91-22792*# College of William and Mary, Williamsburg, VA. Dept. of Computer Science.

**DESIGN OF AN INTELLIGENT INFORMATION
SYSTEM FOR IN-FLIGHT EMERGENCY ASSISTANCE**

STEFAN FEYOCK and STAMOS KARAMOUZIS In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 295-306 May 1991

Avail: CASI HC A03/MF A03

The present research has as its goal the development of AI tools to help flight crews cope with in-flight malfunctions. The relevant tasks in such situations include diagnosis, prognosis, and recovery plan generation. Investigation of the information requirements of these tasks has shown that the determination of paths figures largely: what components or systems are connected to what others, how are they connected, whether connections satisfying certain criteria exist, and a number of related queries. The formulation of such queries frequently requires capabilities of the second-order predicate calculus. An information system is described that features second-order logic capabilities, and is oriented toward efficient formulation and execution of such queries.

Author

N91-22793*# George Mason Univ., Fairfax, VA. Dept. of Physics.

**PRELIMINARY RESULTS OF INVESTIGATIONS INTO
THE USE OF ARTIFICIAL NEURAL NETWORKS FOR
DISCRIMINATING GAS CHROMATOGRAPH MASS
SPECTRA OF REMOTE SAMPLES**

HAROLD A. GELLER (George Mason Univ., Fairfax, V.A.),

EUGENE NORRIS (George Mason Univ., Fairfax, V.A.), and

ARCHIBALD WARNOCK, III (ST Systems Corp., Lanham, MD.) In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 307-316 May 1991 Prepared in cooperation with Research and Data Systems, Inc., Greenbelt, MD
Avail: CASI HC A02/MF A03

Neural networks trained using mass spectra data from the National Institute of Standards and Technology (NIST) are studied. The investigations also included sample data from the gas chromatograph mass spectrometer (GCMS) instrument aboard the Viking Lander, obtained from the National Space Science Data Center. The work performed to date and the preliminary results from the training and testing of neural networks are described. These preliminary results are presented for the purpose of determining the viability of applying artificial neural networks in discriminating mass spectra samples from remote instrumentation such as the Mars Rover Sample Return Mission and the Cassini Probe.

Author

N91-22794*# National Aeronautics and Space Administration. Lyndon B. Johnson Space Center, Houston, TX.

METAMORPHOSES OF ONAV CONSOLE OPERATIONS: FROM PROTOTYPE TO REAL TIME APPLICATION

MALISE MILLIS and LUI WANG In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 317-326 May 1991

Avail: CASI HC A02/MF A03

The ONAV (Onboard Navigation) Expert System is being developed as a real time console assistant to the ONAV flight controller for use in the Mission Control Center at the Johnson Space Center. Currently the entry and rendezvous systems are in verification, and the ascent is being prototyped. To arrive at this stage, from a prototype to real world application, the ONAV project has had to deal with not only AI issues but operating environment issues. The AI issues included the maturity of AI languages and the debugging tools, what is verification, and availability, stability, and the size of the expert pool. The environmental issues included real time data acquisition, hardware stability, and how to achieve acceptance by users and management.

Author

N91-22795*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

LESSONS LEARNED IN THE DEVELOPMENT OF THE STOL INTELLIGENT TUTORING SYSTEM

THOMAS SEAMSTER (Carlow Associates, Inc., Fairfax, VA.), CLIFFORD BAKER (Carlow Associates, Inc., Fairfax, VA.), and TROY AMES In its The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 327-338 May 1991

Avail: CASI HC A03/MF A03

Lessons learned during the development of the NASA Systems Test and Operations Language (STOL) Intelligent Tutoring System (ITS), being developed at NASA Goddard Space Flight Center are presented. The purpose of the intelligent tutor is to train STOL users by adapting tutoring based on inferred student strengths and weaknesses. This system has been under development for over one year and numerous lessons learned have emerged. These observations are presented in three sections, as follows. The first section addresses the methodology employed in the development of the STOL ITS and briefly presents the ITS architecture. The second presents lessons learned, in the areas of: intelligent tutor development;

documentation and reporting; cost and schedule control; and tools and shells effectiveness. The third section presents recommendations which may be considered by other ITS developers, addressing: access, use and selection of subject matter experts; steps involved in ITS development; use of ITS interface design prototypes as part of knowledge engineering; and tools and shells effectiveness.

Author

N91-22796*# Sverdrup Technology, Inc., Brook Park, OH.
SPACE COMMUNICATIONS ARTIFICIAL INTELLIGENCE FOR LINK EVALUATION TERMINAL (SCAILET)

ANOOSH SHAHIDI In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 339-351 May 1991

Avail: CASI HC A03/MF A03

A software application to assist end-users of the Link Evaluation Terminal (LET) for satellite communication is being developed. This software application incorporates artificial intelligence (AI) techniques and will be deployed as an interface to LET. The high burst rate (HBR) LET provides 30 GHz transmitting/20 GHz receiving, 220/110 Mbps capability for wideband communications technology experiments with the Advanced Communications Technology Satellite (ACTS). The HBR LET and ACTS are being developed at the NASA Lewis Research Center. The HBR LET can monitor and evaluate the integrity of the HBR communications uplink and downlink to the ACTS satellite. The uplink HBR transmission is performed by bursting the bit-pattern as a modulated signal to the satellite. By comparing the transmitted bit pattern with the received bit pattern, HBR LET can determine the bit error rate (BER) under various atmospheric conditions. An algorithm for power augmentation is applied to enhance the system's BER performance at reduced signal strength caused by adverse conditions. Programming scripts, defined by the design engineer, set up the HBR LET terminal by programming subsystem devices through IEEE488 interfaces. However, the scripts are difficult to use, require a steep learning curve, are cryptic, and are hard to maintain. The combination of the learning curve and the complexities involved with editing the script files may discourage end-users from utilizing the full capabilities of the HBR LET system. An intelligent assistant component of SCAILET that addresses critical end-user needs in the programming of the HBR LET system as anticipated by its developers is described. A close look is taken at the various steps involved in writing ECM software for a C&P, computer and at how the intelligent assistant improves the HBR LET system and enhances the end-user's ability to perform the experiments.

Author

N91-22797*# Boeing Computer Services Co., Seattle, WA. Advanced Technology Center.

KNOWLEDGE REPOSITORIES FOR MULTIPLE USES

KEITH WILLIAMSON and PATRICIA RIDDLE In NASA. Goddard Space Flight Center, The 1991 Goddard Conference on Space Applications of Artificial Intelligence p 353-367 May 1991

Avail: CASI HC A03/MF A03

In the life cycle of a complex physical device or part, for example, the docking bay door of the Space Station, there are many uses for knowledge about the device or part. The same piece of knowledge might serve several uses. Given the quantity and complexity of the knowledge that must be stored, it is critical to maintain the knowledge in one repository, in one form. At the

same time, because of quantity and complexity of knowledge that must be used in life cycle applications such as cost estimation, re-design, and diagnosis, it is critical to automate such knowledge uses. For each specific use, a knowledge base must be available and must be in a form that promotes the efficient performance of that knowledge base. However, without a single source knowledge repository, the cost of maintaining consistent knowledge between multiple knowledge bases increases dramatically; as facts and descriptions change, they must be updated in each individual knowledge base. A use-neutral representation of a hydraulic system for the F-111 aircraft was developed. The ability to derive portions of four different knowledge bases is demonstrated from this use-neutral representation: one knowledge base is for re-design of the device using a model-based reasoning problem solver; two knowledge bases, at different levels of abstraction, are for diagnosis using a model-based reasoning solver; and one knowledge base is for diagnosis using an associational reasoning problem solver. It was shown how updates issued against the single source use-neutral knowledge repository can be propagated to the underlying knowledge bases.

Author

N92-23356*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE 1992 GODDARD CONFERENCE ON SPACE APPLICATIONS OF ARTIFICIAL INTELLIGENCE

JAMES L. RASH, ed. Washington 1992 251 p Conference held in Greenbelt, MD, 5-6 May 1992

(Contract RTOP 030-09-01-25)

(NASA-CP-3141; REPT-92B00045; NAS 1.55:3141) Avail: CASI HC A12/MF A03

The purpose of this conference is to provide a forum in which current research and development directed at space applications of artificial intelligence can be presented and discussed. The papers fall into the following areas: planning and scheduling, control, fault monitoring/diagnosis and recovery, information management, tools, neural networks, and miscellaneous applications.

N92-23357*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

ARTIFICIAL INTELLIGENCE APPROACH TO PLANNING THE ROBOTIC ASSEMBLY OF LARGE TETRAHEDRAL TRUSS STRUCTURES

LUIZ S. HOMEMDEMELLO In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 3-12 1992

Avail: CASI HC A02/MF A03

An assembly planner for tetrahedral truss structures is presented. To overcome the difficulties due to the large number of parts, the planner exploits the simplicity and uniformity of the shapes of the parts and the regularity of their interconnection. The planning automation is based on the computational formalism known as production system. The global data base consists of a hexagonal grid representation of the truss structure. This representation captures the regularity of tetrahedral truss structures and their multiple hierarchies. It maps into quadratic grids and can be implemented in a computer by using a two-dimensional array data structure. By maintaining the multiple hierarchies explicitly in the model, the choice of a particular hierarchy is only made when needed, thus allowing a more informed decision. Furthermore, testing the preconditions of the production rules is simple because the patterned way in which the struts are interconnected is incorporated into the topology of

the hexagonal grid. A directed graph representation of assembly sequences allows the use of both graph search and backtracking control strategies.

Author

N92-23358*# National Taiwan Univ., Taipei. Dept. of Computer Science and Information Engineering.

SLS-PLAN-IT: A KNOWLEDGE-BASED BLACKBOARD SCHEDULING SYSTEM FOR SPACELAB LIFE SCIENCES MISSIONS

CHENG-YAN KAO (National Taiwan Univ., Taipei.) and SEOK-HUA LEE (General Electric Co., Houston, TX.) In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 13-27 1992

(Contract NAS9-17884)

Avail: CASI HC A03/MF A03

The primary scheduling tool in use during the Spacelab Life Science (SLS-1) planning phase was the operations research (OR) based, tabular form Experiment Scheduling System (ESS) developed by NASA Marshall. PLAN-IT is an artificial intelligence based interactive graphic timeline editor for ESS developed by JPL. The PLAN-IT software was enhanced for use in the scheduling of Spacelab experiments to support the SLS missions. The enhanced software SLS-PLAN-IT System was used to support the real-time reactive scheduling task during the SLS-1 mission. SLS-PLAN-IT is a frame-based blackboard scheduling shell which, from scheduling input, creates resource-requiring event duration objects and resource-usage duration objects. The blackboard structure is to keep track of the effects of event duration objects on the resource usage objects. Various scheduling heuristics are coded in procedural form and can be invoked any time at the user's request. The system architecture is described along with what has been learned with the SLS-PLAN-IT project.

Author

N92-23359*# Space Telescope Science Inst., Baltimore, MD. **DETECTING OPPORTUNITIES FOR PARALLEL OBSERVATIONS ON THE HUBBLE SPACE TELESCOPE**

MICHAEL LUCKS In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 29-44 1992

Avail: CASI HC A03/MF A03

The presence of multiple scientific instruments aboard the Hubble Space Telescope provides opportunities for parallel science, i.e., the simultaneous use of different instruments for different observations. Determining whether candidate observations are suitable for parallel execution depends on numerous criteria (some involving quantitative tradeoffs) that may change frequently. A knowledge based approach is presented for constructing a scoring function to rank candidate pairs of observations for parallel science. In the Parallel Observation Matching System (POMS), spacecraft knowledge and schedulers' preferences are represented using a uniform set of mappings, or knowledge functions. Assessment of parallel science opportunities is achieved via composition of the knowledge functions in a prescribed manner. The knowledge acquisition, and explanation facilities of the system are presented. The methodology is applicable to many other multiple criteria assessment problems.

Author

N92-23360*# Symbiotics, Inc., Cambridge, MA.

COORDINATING COMPLEX PROBLEM-SOLVING AMONG DISTRIBUTED INTELLIGENT AGENTS

RICHARD M. ADLER In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 47-57 1992
(Contract NAS10-11606; NAS10-11763)

Avail: CASI HC A03/MF A03

A process-oriented control model is described for distributed problem solving. The model coordinates the transfer and manipulation of information across independent networked applications, both intelligent and conventional. The model was implemented using SOCIAL, a set of object-oriented tools for distributing computing. Complex sequences of distributed tasks are specified in terms of high level scripts. Scripts are executed by SOCIAL objects called Manager Agents, which realize an intelligent coordination model that routes individual tasks to suitable server applications across the network. These tools are illustrated in a prototype distributed system for decision support of ground operations for NASA's Space Shuttle fleet. Author

N92-23361*# Naval Research Lab., Washington, DC. Interface and Control Systems.

DISTRIBUTED EXPERT SYSTEMS FOR GROUND AND SPACE APPLICATIONS

BRIAN BUCKLEY (Barrios Technology, Inc., Houston, TX.) and LOUIS WHEATCRAFT (Barrios Technology, Inc., Houston, TX.) In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 59-70 1992

Avail: CASI HC A03/MF A03

Presented here is the Spacecraft Command Language (SCL) concept of the unification of ground and space operations using a distributed approach. SCL is a hybrid software environment borrowing from expert system technology, fifth generation language development, and multitasking operating system environments. Examples of potential uses for the system and current distributed applications of SCL are given. Author

N92-23362*# Jet Propulsion Lab., California Inst. of Tech., Pasadena, CA.

EVALUATING MODEL ACCURACY FOR MODEL-BASED REASONING

STEVE CHIEN and JOSEPH RODEN In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 71-76 1992

Avail: CASI HC A02/MF A03

Described here is an approach to automatically assessing the accuracy of various components of a model. In this approach, actual data from the operation of a target system is used to drive statistical measures to evaluate the prediction accuracy of various portions of the model. We describe how these statistical measures of model accuracy can be used in model-based reasoning for monitoring and design. We then describe the application of these techniques to the monitoring and design of the water recovery system of the Environmental Control and Life Support System (ECLSS) of Space Station Freedom. Author

N92-23363*# Loral AeroSys, Seabrook, MD. AN ARCHITECTURE FOR THE DEVELOPMENT OF REAL-TIME FAULT DIAGNOSIS SYSTEMS USING MODEL-BASED REASONING

GARDINER A. HALL, JAMES SCHUETZLE, DAVID LAVALLEE, and UDAY GUPTA In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 77-86 1992

Avail: CASI HC A02/MF A03

Presented here is an architecture for implementing real-time telemetry based diagnostic systems using model-based reasoning. First, we describe Paragon, a knowledge acquisition tool for offline entry and validation of physical system models. Paragon provides domain experts with a structured editing capability to capture the physical component's structure, behavior, and causal relationships. We next describe the architecture of the run time diagnostic system. The diagnostic system, written entirely in Ada, uses the behavioral model developed offline by Paragon to simulate expected component states as reflected in the telemetry stream. The diagnostic algorithm traces causal relationships contained within the model to isolate system faults. Since the diagnostic process relies exclusively on the behavioral model and is implemented without the use of heuristic rules, it can be used to isolate unpredicted faults in a wide variety of systems. Finally, we discuss the implementation of a prototype system constructed using this technique for diagnosing faults in a science instrument. The prototype demonstrates the use of model-based reasoning to develop maintainable systems with greater diagnostic capabilities at a lower cost. Author

N92-23364*# General Electric Co., Philadelphia, PA. Military and Data Systems Organization.

THE ACHIEVEMENT OF SPACECRAFT AUTONOMY THROUGH THE THEMATIC APPLICATION OF MULTIPLE COOPERATING INTELLIGENT AGENTS

PHILIP J. ROSSOMANDO In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 87-103 1992

Avail: CASI HC A03/MF A03

A description is given of UNICORN, a prototype system developed for the purpose of investigating artificial intelligence (AI) concepts supporting spacecraft autonomy. UNICORN employs thematic reasoning, of the type first described by Rodger Schank of Northwestern University, to allow the context-sensitive control of multiple intelligent agents within a blackboard based environment. In its domain of application, UNICORN demonstrates the ability to reason teleologically with focused knowledge. Also presented are some of the lessons learned as a result of this effort. These lessons apply to any effort wherein system level autonomy is the objective. Author

N92-23365*# National Aeronautics and Space Administration. Lewis Research Center, Cleveland, OH.

INTELLIGENT FAULT ISOLATION AND DIAGNOSIS FOR COMMUNICATION SATELLITE SYSTEMS

DONALD P. TALLO (Akron Univ., OH.), JOHN DURKIN (Akron Univ., OH.), and EDWARD J. PETRIK In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 105-120 1992

Avail: CASI HC A03/MF A03

Discussed here is a prototype diagnosis expert system to provide the Advanced Communication Technology Satellite (ACTS) System with autonomous diagnosis capability. The system, the Fault Isolation and Diagnosis EXpert (FIDEX) system, is a frame-based system that uses hierarchical structures to represent such items as the satellite's subsystems, components, sensors, and fault states. This overall frame architecture integrates the hierarchical structures into a lattice that provides a flexible representation scheme and facilitates system maintenance. FIDEX uses an inexact reasoning technique based

on the incrementally acquired evidence approach developed by Shortliffe. The system is designed with a primitive learning ability through which it maintains a record of past diagnosis studies.

Author

N92-23366*# Colorado Univ., Colorado Springs, CO. Dept. of Computer Science.

FEATURE DETECTION IN SATELLITE IMAGES USING NEURAL NETWORK TECHNOLOGY

MARIJKE F. AUGUSTEIJN and ARTURO S. DIMALANTA In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 123-136 1992 Sponsored in part by CTA, Inc.

Avail: CASI HC A03/MF A03

A feasibility study of automated classification of satellite images is described. Satellite images were characterized by the textures they contain. In particular, the detection of cloud textures was investigated. The method of second-order gray level statistics, using co-occurrence matrices, was applied to extract feature vectors from image segments. Neural network technology was employed to classify these feature vectors. The cascade-correlation architecture was successfully used as a classifier. The use of a Kohonen network was also investigated but this architecture could not reliably classify the feature vectors due to the complicated structure of the classification problem. The best results were obtained when data from different spectral bands were fused.

Author

N92-23367*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

DESIGN OF NEURAL NETWORKS FOR CLASSIFICATION OF REMOTELY SENSED IMAGERY

SAMIR R. CHETTRI (Hughes STX, Inc., Greenbelt, MD.), ROBERT F. CROMP (National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.), and MARK BIRMINGHAM (Princeton Univ., NJ.) In its The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 137-150 1992

Avail: CASI HC A03/MF A03

Classification accuracies of a backpropagation neural network are discussed and compared with a maximum likelihood classifier (MLC) with multivariate normal class models. We have found that, because of its nonparametric nature, the neural network outperforms the MLC in this area. In addition, we discuss techniques for constructing optimal neural nets on parallel hardware like the MasPar MP-1 currently at GSFC. Other important discussions are centered around training and classification times of the two methods, and sensitivity to the training data. Finally, we discuss future work in the area of classification and neural nets.

Author

N92-23368*# Computer Application Systems, Signal Mountain, TN.

IMPROVED IMAGE CLASSIFICATION WITH NEURAL NETWORKS BY FUSING MULTISPECTRAL SIGNATURES WITH TOPOLOGICAL DATA

CRAIG HARSTON and CHRIS SCHUMACHER In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 151-158 1992 (Contract NAS13-435)

Avail: CASI HC A02/MF A03

Automated schemes are needed to classify multispectral remotely sensed data. Human intelligence is often required to correctly interpret images from satellites and aircraft.

Humans succeed because they use various types of cues about a scene to accurately define the contents of the image. Consequently, it follows that computer techniques that integrate and use different types of information would perform better than single source approaches. This research illustrated that multispectral signatures and topographical information could be used in concert. Significantly, this dual source tactic classified a remotely sensed image better than the multispectral classification alone. These classifications were accomplished by fusing spectral signatures with topographical information using neural network technology. A neural network was trained to classify Landsat multispectral signatures. A file of georeferenced ground truth classifications were used as the training criterion. The network was trained to classify urban, agriculture, range, and forest with an accuracy of 65.7 percent. Another neural network was programmed and trained to fuse these multispectral signature results with a file of georeferenced altitude data. This topological file contained 10 levels of elevations. When this nonspectral elevation information was fused with the spectral signatures, the classifications were improved to 73.7 and 75.7 percent. Author

N92-23369*# Loyola Univ., New Orleans, LA. Dept. of Mathematical Sciences.

IMPROVED INTERPRETATION OF SATELLITE ALTIMETER DATA USING GENETIC ALGORITHMS

KENNETH MESSA (Princeton Univ., NJ.) and MATTHEW LYBANON (Naval Oceanographic and Atmospheric Research Lab., Bay Saint Louis, MS.) In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 159-166 1992 Sponsored in part by Navy

Avail: CASI HC A02/MF A03

Genetic algorithms (GA) are optimization techniques that are based on the mechanics of evolution and natural selection. They take advantage of the power of cumulative selection, in which successive incremental improvements in a solution structure become the basis for continued development. A GA is an iterative procedure that maintains a 'population' of 'organisms' (candidate solutions). Through successive 'generations' (iterations) the population as a whole improves in simulation of Darwin's 'survival of the fittest'. GA's have been shown to be successful where noise significantly reduces the ability of other search techniques to work effectively. Satellite altimetry provides useful information about oceanographic phenomena. It provides rapid global coverage of the oceans and is not as severely hampered by cloud cover as infrared imagery. Despite these and other benefits, several factors lead to significant difficulty in interpretation. The GA approach to the improved interpretation of satellite data involves the representation of the ocean surface model as a string of parameters or coefficients from the model. The GA searches in parallel, a population of such representations (organisms) to obtain the individual that is best suited to 'survive', that is, the fittest as measured with respect to some 'fitness' function. The fittest organism is the one that best represents the ocean surface model with respect to the altimeter data.

Author

N92-23370*# Teledyne Brown Engineering, Huntsville, AL. Space Operations Dept.

THE USE OF ARTIFICIAL INTELLIGENCE TECHNIQUES TO IMPROVE THE MULTIPLE PAYLOAD INTEGRATION PROCESS

DANNIE E. CUTTS and BRIAN K. WIDGREN In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on

Space Applications of Artificial Intelligence p 169-179 1992
Avail: CASI HC A03/MF A03

A maximum return of science and products with a minimum expenditure of time and resources is a major goal of mission payload integration. A critical component then, in successful mission payload integration is the acquisition and analysis of experiment requirements from the principal investigator and payload element developer teams. One effort to use artificial intelligence techniques to improve the acquisition and analysis of experiment requirements within the payload integration process is described.

Author

N92-23371*# George Mason Univ., Fairfax, VA. Dept. of Information and Software Systems Engineering.

KNOWLEDGE-BASED APPROACH FOR GENERATING TARGET SYSTEM SPECIFICATIONS FROM A DOMAIN MODEL

HASSAN GOMAA, LARRY KERSCHBERG, and VIJAYAN SUGUMARAN In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 181-194 1992

Avail: CASI HC A03/MF A03

Several institutions in industry and academia are pursuing research efforts in domain modeling to address unresolved issues in software reuse. To demonstrate the concepts of domain modeling and software reuse, a prototype software engineering environment is being developed at George Mason University to support the creation of domain models and the generation of target system specifications. This prototype environment, which is application domain independent, consists of an integrated set of commercial off-the-shelf software tools and custom-developed software tools. This paper describes the knowledge-based tool that was developed as part of the environment to generate target system specifications from a domain model.

Author

N92-23372*# University of Central Florida, Orlando, FL. Dept. of Computer Science.

COMBINING FACTUAL AND HEURISTIC KNOWLEDGE IN KNOWLEDGE ACQUISITION

FERNANDO GOMEZ (University of Central Florida, Orlando.), RICHARD HULL (University of Central Florida, Orlando.), CLARK KARR (University of Central Florida, Orlando.), BRUCE HOSKEN (University of Central Florida, Orlando.), and WILLIAM VERHAGEN (Grumman Aerospace Corp., Bethpage, NY.) In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 195-209 1992

(Contract NAG10-0058)

Avail: CASI HC A03/MF A03

A knowledge acquisition technique that combines heuristic and factual knowledge represented as two hierarchies is described. These ideas were applied to the construction of a knowledge acquisition interface to the Expert System Analyst (OPERA). The goal of OPERA is to improve the operations support of the computer network in the space shuttle launch processing system. The knowledge acquisition bottleneck lies in gathering knowledge from human experts and transferring it to OPERA. OPERA's knowledge acquisition problem is approached as a classification problem-solving task, combining this approach with the use of factual knowledge about the domain. The interface was implemented in a Symbolics workstation making heavy use of windows, pull-down menus, and other user-friendly devices.

Author

N92-23373*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

A SPATIAL DATA HANDLING SYSTEM FOR RETRIEVAL OF IMAGES BY UNRESTRICTED REGIONS OF USER INTEREST

ERIK DORFMAN (Hughes STX, Inc., Greenbelt, MD.) and ROBERT F. CROMP In its The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 213-230 1992
Avail: CASI HC A03/MF A03

The Intelligent Data Management (IDM) project at NASA/Goddard Space Flight Center has prototyped an Intelligent Information Fusion System (IIFS), which automatically ingests metadata from remote sensor observations into a large catalog which is directly queryable by end-users. The greatest challenge in the implementation of this catalog was supporting spatially-driven searches, where the user has a possible complex region of interest and wishes to recover those images that overlap all or simply a part of that region. A spatial data management system is described, which is capable of storing and retrieving records of image data regardless of their source. This system was designed and implemented as part of the IIFS catalog. A new data structure, called a hypercylinder, is central to the design. The hypercylinder is specifically tailored for data distributed over the surface of a sphere, such as satellite observations of the Earth or space. Operations on the hypercylinder are regulated by two expert systems. The first governs the ingest of new metadata records, and maintains the efficiency of the data structure as it grows. The second translates, plans, and executes users' spatial queries, performing incremental optimization as partial query results are returned.

Author

N92-23374*# Argonne National Lab., IL. Environmental Assessment and Information Sciences Div.

DATA EXPLORATION SYSTEMS FOR DATABASES

RICHARD J. GREENE and CHRISTOPHER HIELD In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 231-244 1992

Avail: CASI HC A03/MF A03

Data exploration systems apply machine learning techniques, multivariate statistical methods, information theory, and database theory to databases to identify significant relationships among the data and summarize information. The result of applying data exploration systems should be a better understanding of the structure of the data and a perspective of the data enabling an analyst to form hypotheses for interpreting the data. This paper argues that data exploration systems need a minimum amount of domain knowledge to guide both the statistical strategy and the interpretation of the resulting patterns discovered by these systems.

Author

N92-23375*# Loyola Univ., New Orleans, LA. Mathematical Sciences.

LOGIC PROGRAMMING AND METADATA SPECIFICATIONS

ANTONIO M. LOPEZ, JR. (Loyola Univ., New Orleans, LA.) and MARGUERITE E. SAACKS (Xavier Univ. of Louisiana, New Orleans.) In NASA. Goddard Space Flight Center, The 1992 Goddard Conference on Space Applications of Artificial Intelligence p 245-253 1992 Sponsored by NASA. Stennis Space Center

Avail: CASI HC A02/MF A03

Artificial intelligence (AI) ideas and techniques are critical to the development of intelligent information systems that

will be used to collect, manipulate, and retrieve the vast amounts of space data produced by 'Missions to Planet Earth.' Natural language processing, inference, and expert systems are at the core of this space application of AI. This paper presents logic programming as an AI tool that can support inference (the ability to draw conclusions from a set of complicated and interrelated facts). It reports on the use of logic programming in the study of metadata specifications for a small problem domain of airborne sensors, and the dataset characteristics and pointers that are needed for data access.

Author

N93-25961*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE 1993 GODDARD CONFERENCE ON SPACE APPLICATIONS OF ARTIFICIAL INTELLIGENCE
CARL F. HOSTETTER, ed. Washington 1993 280 p
Conference held in Greenbelt, MD, 10-13 May 1993
(NASA-CP-3200; NAS 1.55:3200)

Avail: CASI HC A13/MF A03

This publication comprises the papers presented at the 1993 Goddard Conference on Space Applications of Artificial Intelligence held at the NASA/Goddard Space Flight Center, Greenbelt, MD on May 10-13, 1993. The purpose of this annual conference is to provide a forum in which current research and development directed at space applications of artificial intelligence can be presented and discussed.

N93-25962*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

USING AI/EXPERT SYSTEM TECHNOLOGY TO AUTOMATE PLANNING AND REPLANNING FOR THE HST SERVICING MISSIONS

L. BOGOVICH (Bendix Field Engineering Corp., Seabrook, MD.), J JOHNSON (Bendix Field Engineering Corp., Seabrook, MD.), A. TUCHMAN (Bendix Field Engineering Corp., Seabrook, MD.), D. MCLEAN (Bendix Field Engineering Corp., Seabrook, MD.), B. PAGE (Bendix Field Engineering Corp., Seabrook, MD.), A. KISPERT (Bendix Field Engineering Corp., Seabrook, MD.), C. BURKHARDT (Bendix Field Engineering Corp., Seabrook, MD.), R. LITTLEFIELD (Bendix Field Engineering Corp., Seabrook, MD.), and W. POTTER In its The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 3-10 1993

(Contract NAS5-27772)

Avail: CASI HC A02/MF A03

This paper describes a knowledge-based system that has been developed to automate planning and scheduling for the Hubble Space Telescope (HST) Servicing Missions. This new system is the Servicing Mission Planning and Replanning Tool (SM/PART). SM/PART has been delivered to the HST Flight Operations Team (FOT) at Goddard Space Flight Center (GSFC) where it is being used to build integrated time lines and command plans to control the activities of the HST, Shuttle, Crew and ground systems for the next HST Servicing Mission. SM/PART reuses and extends AI/expert system technology from Interactive Experimenter Planning System (IEPS) systems to build or rebuild time lines and command plans more rapidly than was possible for previous missions where they were built manually. This capability provides an important safety factor for the HST, Shuttle and Crew in case unexpected events occur during the mission.

Author

N93-25963*# Honeywell, Inc., Minneapolis, MN.
KNOWLEDGE-BASED CONTROL FOR ROBOT SELF-

LOCALIZATION

BONNIE KATHLEEN HOLT BENNETT In NASA. Goddard Space Flight Center, The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 13-27 1993 Sponsored in part by DARPA Prepared in cooperation with Saint Thomas Coll., Saint Paul, MN
(Contract NSF IRI-91-96146)

Avail: CASI HC A03/MF A03

Autonomous robot systems are being proposed for a variety of missions including the Mars rover/sample return mission. Prior to any other mission objectives being met, an autonomous robot must be able to determine its own location. This will be especially challenging because location sensors like GPS, which are available on Earth, will not be useful, nor will INS sensors because their drift is too large. Another approach to self-localization is required. In this paper, we describe a novel approach to localization by applying a problem solving methodology. The term 'problem solving' implies a computational technique based on logical representational and control steps. In this research, these steps are derived from observing experts solving localization problems. The objective is not specifically to simulate human expertise but rather to apply its techniques where appropriate for computational systems. In doing this, we describe a model for solving the problem and a system built on that model, called localization control and logic expert (LOCATE), which is a demonstration of concept for the approach and the model. The results of this work represent the first successful solution to high-level control aspects of the localization problem.

Author

N93-25964*# Institute for Space and Terrestrial Science, North York (Ontario).

PARAMETRIC MOTION CONTROL OF ROBOTIC ARMS: A BIOLOGICALLY BASED APPROACH USING NEURAL NETWORKS

O. BOCK, G. M. T. D'ELEUTERIO (Toronto Univ., Ontario.), J. LIPITKAS (Toronto Univ., Ontario.), and J. J. GRODSKI (Defence and Civil Inst. of Environmental Medicine, Toronto, Ontario.) In NASA. Goddard Space Flight Center, The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 29-37 1993

(Contract W7711-0-7118)

Avail: CASI HC A02/MF A03

A neural network based system is presented which is able to generate point-to-point movements of robotic manipulators. The foundation of this approach is the use of prototypical control torque signals which are defined by a set of parameters. The parameter set is used for scaling and shaping of these prototypical torque signals to effect a desired outcome of the system. This approach is based on neurophysiological findings that the central nervous system stores generalized cognitive representations of movements called synergies, schemas, or motor programs. It has been proposed that these motor programs may be stored as torque-time functions in central pattern generators which can be scaled with appropriate time and magnitude parameters. The central pattern generators use these parameters to generate stereotypical torque-time profiles, which are then sent to the joint actuators. Hence, only a small number of parameters need to be determined for each point-to-point movement instead of the entire torque-time trajectory. This same principle is implemented for controlling the joint torques of robotic manipulators where a neural network is used to identify the relationship between the task requirements and the torque parameters. Movements are specified by the initial robot position

in joint coordinates and the desired final end-effector position in Cartesian coordinates. This information is provided to the neural network which calculates six torque parameters for a two-link system. The prototypical torque profiles (one per joint) are then scaled by those parameters. After appropriate training of the network, our parametric control design allowed the reproduction of a trained set of movements with relatively high accuracy, and the production of previously untrained movements with comparable accuracy. We conclude that our approach was successful in discriminating between trained movements and in generalizing to untrained movements.

Author

**N93-25965*# Allied-Signal Aerospace Co., Columbia, MD.
DETECTION OF BEARING FAILURE IN MECHANICAL
DEVICES USING NEURAL NETWORKS**

RICHARD A. BURNE, PAUL F. PAYER, R. PAUL GORMAN,
and DAN T. HORAK In NASA. Goddard Space Flight Center,
The 1993 Goddard Conference on Space Applications of
Artificial Intelligence p 39-45 1993

Avail: CASI HC A02/MF A03

We present a novel time-domain method for the detection of faulty bearings that has direct applicability to monitoring the health of the turbo pumps on the Space Shuttle Main Engine. A feed-forward neural network was trained to detect modelled roller bearing faults on the basis of the periodicity of impact pulse trains. The network's performance was dependent upon the number of pulses in the network's input window and the signal-to-noise ratio of the input signal. To test the model's validity, we fit the model's parameters to an actual vibration signal generated by a faulty roller element bearing and applied the network trained on this model to detect faults in actual vibration data. When this network was tested on the actual vibration data, it correctly identified the vibration signal as a fault condition 76 percent of the time.

Author

**N93-25966*# Boeing Defense and Space Group, Huntsville,
AL. Advanced Computing Lab.**

**MACHINE LEARNING TECHNIQUES FOR FAULT
ISOLATION AND SENSOR PLACEMENT**

JAMES R. CARNES and DOUGLAS H. FISHER (Vanderbilt
Univ., Nashville, TN.) In NASA. Goddard Space Flight Center,
The 1993 Goddard Conference on Space Applications of
Artificial Intelligence p 47-58 1993

Avail: CASI HC A03/MF A03

Fault isolation and sensor placement are vital for monitoring and diagnosis. A sensor conveys information about a system's state that guides troubleshooting if problems arise. We are using machine learning methods to uncover behavioral patterns over snapshots of system simulations that will aid fault isolation and sensor placement, with an eye towards minimality, fault coverage, and noise tolerance.

Author

**N93-25967*# Harris Corp., Melbourne, FL.
ADAPTIVE LASER LINK RECONFIGURATION USING
CONSTRAINT PROPAGATION**

M. S. CRONE, P. M. JULICH, and L. M. COOK In NASA.
Goddard Space Flight Center, The 1993 Goddard Conference on
Space Applications of Artificial Intelligence p 59-73 1993
(Contract F30602-89-D-0096)

Avail: CASI HC A03/MF A03

This paper describes Harris AI research performed on the Adaptive Link Reconfiguration (ALR) study for Rome Lab, and focuses on the application of constraint propagation to the problem of link reconfiguration for the proposed space based

Strategic Defense System (SDS) Brilliant Pebbles (BP) communications system. According to the concept of operations at the time of the study, laser communications will exist between BP's and to ground entry points. Long-term links typical of RF transmission will not exist. This study addressed an initial implementation of BP's based on the Global Protection Against Limited Strikes (GPALS) SDI mission. The number of satellites and rings studied was representative of this problem. An orbital dynamics program was used to generate line-of-site data for the modeled architecture. This was input into a discrete event simulation implemented in the Harris developed COnstraint Propagation Expert System (COPES) Shell, developed initially on the Rome Lab BM/C3 study. Using a model of the network and several heuristics, the COPES shell was used to develop the Heuristic Adaptive Link Ordering (HALO) Algorithm to rank and order potential laser links according to probability of communication. A reduced set of links based on this ranking would then be used by a routing algorithm to select the next hop. This paper includes an overview of Constraint Propagation as an Artificial Intelligence technique and its embodiment in the COPES shell. It describes the design and implementation of both the simulation of the GPALS BP network and the HALO algorithm in COPES. This is described using a 59 Data Flow Diagram, State Transition Diagrams, and Structured English PDL. It describes a laser communications model and the heuristics involved in rank-ordering the potential communication links. The generation of simulation data is described along with its interface via COPES to the Harris developed View Net graphical tool for visual analysis of communications networks. Conclusions are presented, including a graphical analysis of results depicting the ordered set of links versus the set of all possible links based on the computed Bit Error Rate (BER). Finally, future research is discussed which includes enhancements to the HALO algorithm, network simulation, and the addition of an intelligent routing algorithm for BP.

Author

N93-25968*# Houston Univ., TX.

**AN ARCHITECTURE FOR OBJECT-ORIENTED
INTELLIGENT CONTROL OF POWER SYSTEMS IN
SPACE**

SVEN G. HOLMQUIST, PRAKASH JAYARAM, and BEN H.
JANSEN In NASA. Goddard Space Flight Center, The 1993
Goddard Conference on Space Applications of Artificial
Intelligence p 75-82 1993

Avail: CASI HC A02/MF A03

A control system for autonomous distribution and control of electrical power during space missions is being developed. This system should free the astronauts from localizing faults and reconfiguring loads if problems with the power distribution and generation components occur. The control system uses an object-oriented simulation model of the power system and first principle knowledge to detect, identify, and isolate faults. Each power system component is represented as a separate object with knowledge of its normal behavior. The reasoning process takes place at three different levels of abstraction: the Physical Component Model (PCM) level, the Electrical Equivalent Model (EEM) level, and the Functional System Model (FSM) level, with the PCM the lowest level of abstraction and the FSM the highest. At the EEM level the power system components are reasoned about as their electrical equivalents, e.g., a resistive load is thought of as a resistor. However, at the PCM level detailed knowledge about the component's specific characteristics is taken into account. The

FSM level models the system at the subsystem level, a level appropriate for reconfiguration and scheduling. The control system operates in two modes, a reactive and a proactive mode, simultaneously. In the reactive mode the control system receives measurement data from the power system and compares these values with values determined through simulation to detect the existence of a fault. The nature of the fault is then identified through a model-based reasoning process using mainly the EEM. Compound component models are constructed at the EEM level and used in the fault identification process. In the proactive mode the reasoning takes place at the PCM level. Individual components determine their future health status using a physical model and measured historical data. In case changes in the health status seem imminent the component warns the control system about its impending failure. The fault isolation process uses the FSM level for its reasoning base.

Author

N93-25969*# College of William and Mary, Williamsburg, VA.
Dept. of Computer Science.

THE USE OF MULTIPLE MODELS IN CASE-BASED DIAGNOSIS

STAMOS T. KARAMOUZIS and STEFAN FEYOCK In NASA. Goddard Space Flight Center, The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 83-90 1993

(Contract NCC1-159)

Avail: CASI HC A02/MF A03

The work described in this paper has as its goal the integration of a number of reasoning techniques into a unified intelligent information system that will aid flight crews with malfunction diagnosis and prognostication. One of these approaches involves using the extensive archive of information contained in aircraft accident reports along with various models of the aircraft as the basis for case-based reasoning about malfunctions. Case-based reasoning draws conclusions on the basis of similarities between the present situation and prior experience. We maintain that the ability of a CBR program to reason about physical systems is significantly enhanced by the addition to the CBR program of various models. This paper describes the diagnostic concepts implemented in a prototypical case based reasoner that operates in the domain of in-flight fault diagnosis, the various models used in conjunction with the reasoner's CBR component, and results from a preliminary evaluation.

Author

N93-25970*# New South Wales Univ., Kensington (Australia).
Dept. of Computer Science and Engineering.

AN AUTONOMOUS SATELLITE ARCHITECTURE INTEGRATING DELIBERATIVE REASONING AND BEHAVIOURAL INTELLIGENCE

CRAIG A. LINDLEY In NASA. Goddard Space Flight Center, The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 91-106 1993

Avail: CASI HC A03/MF A03

This paper describes a method for the design of autonomous spacecraft, based upon behavioral approaches to intelligent robotics. First, a number of previous spacecraft automation projects are reviewed. A methodology for the design of autonomous spacecraft is then presented, drawing upon both the European Space Agency technological center (ESTEC) automation and robotics methodology and the subsumption architecture for autonomous robots. A layered competency model for autonomous orbital spacecraft is proposed. A simple example of low level competencies and their interaction is presented in

order to illustrate the methodology. Finally, the general principles adopted for the control hardware design of the AUSTRALIS-1 spacecraft are described. This system will provide an orbital experimental platform for spacecraft autonomy studies, supporting the exploration of different logical control models, different computational metaphors within the behavioral control framework, and different mappings from the logical control model to its physical implementation.

Author

N93-25971*# National Oceanic and Atmospheric Administration, Camp Springs, MD.

ANOMALOUS EVENT DIAGNOSIS FOR ENVIRONMENTAL SATELLITE SYSTEMS

BRUCE H. RAMSAY In NASA. Goddard Space Flight Center, The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 107-116 1993

Avail: CASI HC A02/MF A03

The National Oceanic and Atmospheric Administration's (NOAA) National Environmental Satellite, Data, and Information Service (NESDIS) is responsible for the operation of the NOAA geostationary and polar orbiting satellites. NESDIS provides a wide array of operational meteorological and oceanographic products and services and operates various computer and communication systems on a 24-hour, seven days per week schedule. The Anomaly Reporting System contains a database of anomalous events regarding the operations of the Geostationary Operational Environmental Satellite (GOES), communication, or computer systems that have degraded or caused the loss of GOES imagery. Data is currently entered manually via an automated query user interface. There are 21 possible symptoms (e.g., No Data), and 73 possible causes (e.g., Sectorizer - World Weather Building) of an anomalous event. The determination of an event's cause(s) is made by the on-duty computer operator, who enters the event in a paper based daily log, and by the analyst entering the data into the reporting system. The determination of the event's cause(s) impacts both the operational status of these systems, and the performance evaluation of the on-site computer and communication operations contractor.

Author

N93-25972*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE PROBABILISTIC NEURAL NETWORK ARCHITECTURE FOR HIGH SPEED CLASSIFICATION OF REMOTELY SENSED IMAGERY

SAMIR R. CHETTRI and ROBERT F. CROMP In its The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 119-132 1993

Avail: CASI HC A03/MF A03

In this paper we discuss a neural network architecture (the Probabilistic Neural Net or the PNN) that, to the best of our knowledge, has not previously been applied to remotely sensed data. The PNN is a supervised non-parametric classification algorithm as opposed to the Gaussian maximum likelihood classifier (GMLC). The PNN works by fitting a Gaussian kernel to each training point. The width of the Gaussian is controlled by a tuning parameter called the window width. If very small widths are used, the method is equivalent to the nearest neighbor method. For large windows, the PNN behaves like the GMLC. The basic implementation of the PNN requires no training time at all. In this respect it is far better than the commonly used backpropagation neural network which can be shown to take O(N⁶) time for training where N is the dimensionality of the input vector. In addition the PNN can be implemented in a feed

forward mode in hardware. The disadvantage of the PNN is that it requires all the training data to be stored. Some solutions to this problem are discussed in the paper. Finally, we discuss the accuracy of the PNN with respect to the GMLC and the backpropagation neural network (BPNN). The PNN is shown to be better than GMLC and not as good as the BPNN with regards to classification accuracy.

Author

N93-25973*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

IMAGE ANALYSIS BY INTEGRATION OF DISPARATE INFORMATION

JACQUELINE LEMOIGNE In its The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 133-144 1993

Avail: CASI HC A03/MF A03

Image analysis often starts with some preliminary segmentation which provides a representation of the scene needed for further interpretation. Segmentation can be performed in several ways, which are categorized as pixel based, edge-based, and region-based. Each of these approaches are affected differently by various factors, and the final result may be improved by integrating several or all of these methods, thus taking advantage of their complementary nature. In this paper, we propose an approach that integrates pixel-based and edge-based results by utilizing an iterative relaxation technique. This approach has been implemented on a massively parallel computer and tested on some remotely sensed imagery from the Landsat-Thematic Mapper (TM) sensor.

Author

N93-25974*# Caelum Research Corp., Silver Spring, MD.

DATA FUSION WITH ARTIFICIAL NEURAL NETWORKS (ANN) FOR CLASSIFICATION OF EARTH SURFACE FROM MICROWAVE SATELLITE MEASUREMENTS

Y. M. FLEMING LURE, NORMAN C. GRODY (National Oceanic and Atmospheric Administration, Washington, DC.), Y. S. PETER CHIOU, and H. Y. MICHAEL YEH In NASA. Goddard Space Flight Center, The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 145-154 1993 (Contract F30602-89-C-0130)

Avail: CASI HC A02/MF A03

A data fusion system with artificial neural networks (ANN) is used for fast and accurate classification of five earth surface conditions and surface changes, based on seven SSMI multichannel microwave satellite measurements. The measurements include brightness temperatures at 19, 22, 37, and 85 GHz at both H and V polarizations (only V at 22 GHz). The seven channel measurements are processed through a convolution computation such that all measurements are located at same grid. Five surface classes including non-scattering surface, precipitation over land, over ocean, snow, and desert are identified from ground-truth observations. The system processes sensory data in three consecutive phases: (1) pre-processing to extract feature vectors and enhance separability among detected classes; (2) preliminary classification of Earth surface patterns using two separate and parallelly acting classifiers: back-propagation neural network and binary decision tree classifiers; and (3) data fusion of results from preliminary classifiers to obtain the optimal performance in overall classification. Both the binary decision tree classifier and the fusion processing centers are implemented by neural network architectures. The fusion system configuration is a hierarchical neural network architecture, in which each functional neural net will handle

different processing phases in a pipelined fashion. There is a total of around 13,500 samples for this analysis, of which 4 percent are used as the training set and 96 percent as the testing set. After training, this classification system is able to bring up the detection accuracy to 94 percent compared with 88 percent for back-propagation artificial neural networks and 80 percent for binary decision tree classifiers. The neural network data fusion classification is currently under progress to be integrated in an image processing system at NOAA and to be implemented in a prototype of a massively parallel and dynamically reconfigurable Modular Neural Ring (MNR).

Author

N93-25975*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

NEURAL NETWORKS FOR ATMOSPHERIC RETRIEVALS

HOWARD E. MOTTELER, J. A. GUALTIERI, L. LARRABEE STROW (Maryland Univ. Baltimore County, Catonsville.), and LARRY McMILLIN (National Oceanic and Atmospheric Administration, Camp Springs, MD.) In its The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 155-167 1993

Avail: CASI HC A03/MF A03

We use neural networks to perform retrievals of temperature and water fractions from simulated clear air radiances for the Atmospheric Infrared Sounder (AIRS). Neural networks allow us to make effective use of the large AIRS channel set, and give good performance with noisy input. We retrieve surface temperature, air temperature at 64 distinct pressure levels, and water fractions at 50 distinct pressure levels. Using 728 temperature and surface sensitive channels, the RMS error for temperature retrievals with 0.2K input noise is 1.2K. Using 586 water and temperature sensitive channels, the mean error with 0.2K input noise is 16 percent. Our implementation of backpropagation training for neural networks on the 16,000-processor MasPar MP-1 runs at a rate of 90 million weight updates per second, and allows us to train large networks in a reasonable amount of time. Once trained, the network can be used to perform retrievals quickly on a workstation of moderate power.

Author

N93-25976*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

CLASSIFYING MULTISPECTRAL DATA BY NEURAL NETWORKS

BRIAN A. TELFER (Naval Surface Warfare Center, Silver Spring, MD.), HAROLD H. SZU (Naval Surface Warfare Center, Silver Spring, MD.), and RICHARD K. KIANG In its The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 169-177 1993

Avail: CASI HC A02/MF A03

Several energy functions for synthesizing neural networks are tested on 2-D synthetic data and on Landsat-4 Thematic Mapper data. These new energy functions, designed specifically for minimizing misclassification error, in some cases yield significant improvements in classification accuracy over the standard least mean squares energy function. In addition to operating on networks with one output unit per class, a new energy function is tested for binary encoded outputs, which result in smaller network sizes. The Thematic Mapper data (four bands were used) is classified on a single pixel basis, to provide a starting benchmark against which further improvements will be measured. Improvements are underway to make use of both subpixel and superpixel (i.e. contextual or neighborhood)

information in tile processing. For single pixel classification, the best neural network result is 78.7 percent, compared with 71.7 percent for a classical nearest neighbor classifier. The 78.7 percent result also improves on several earlier neural network results on this data.

Author

N93-25977*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE WORKPLACE DISTRIBUTED PROCESSING ENVIRONMENT

TROY AMES and SCOTT HENDERSON (Computer Technology Associates, Inc., Rockville, MD.) In its The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 181-188 1993

Avail: CASI HC A02/MF A03

Real time control problems require robust, high performance solutions. Distributed computing can offer high performance through parallelism and robustness through redundancy. Unfortunately, implementing distributed systems with these characteristics places a significant burden on the applications programmers. Goddard Code 522 has developed WorkPlace to alleviate this burden. WorkPlace is a small, portable, embeddable network interface which automates message routing, failure detection, and re-configuration in response to failures in distributed systems. This paper describes the design and use of WorkPlace, and its application in the construction of a distributed blackboard system.

Author

N93-25978*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

MODEL-BASED REASONING FOR SYSTEM AND SOFTWARE ENGINEERING: THE KNOWLEDGE FROM PICTURES (KFP) ENVIRONMENT

SYDNEY BAILIN (Computer Technology Associates, Inc., Rockville, MD.), FRANK PATERRA (Computer Technology Associates, Inc., Rockville, MD.), SCOTT HENDERSON (Computer Technology Associates, Inc., Rockville, MD.), and WALT TRUSzkowski In its The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 189-200 1993

Avail: CASI HC A03/MF A03

This paper presents a discussion of current work in the area of graphical modeling and model-based reasoning being undertaken by the Automation Technology Section, Code 522.3, at Goddard. The work was initially motivated by the growing realization that the knowledge acquisition process was a major bottleneck in the generation of fault detection, isolation, and repair (FDIR) systems for application in automated Mission Operations. As with most research activities this work started out with a simple objective: to develop a proof-of-concept system demonstrating that a draft rule-base for a FDIR system could be automatically realized by reasoning from a graphical representation of the system to be monitored. This work was called Knowledge From Pictures (KFP) (Truszkowski et. al. 1992). As the work has successfully progressed the KFP tool has become an environment populated by a set of tools that support a more comprehensive approach to model-based reasoning. This paper continues by giving an overview of the graphical modeling objectives of the work, describing the three tools that now populate the KFP environment, briefly presenting a discussion of related work in the field, and by indicating future directions for the KFP environment.

Author

N93-25979*# University of Central Florida, Orlando, FL. Dept. of Computer Science.

INFERRING HEURISTIC CLASSIFICATION HIERARCHIES FROM NATURAL LANGUAGE INPUT

RICHARD HULL and FERNANDO GOMEZ In NASA. Goddard Space Flight Center, The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 201-215 1993 (Contract NAG10-0058)

Avail: CASI HC A03/MF A03

A methodology for inferring hierarchies representing heuristic knowledge about the check out, control, and monitoring sub-system (CCMS) of the space shuttle launch processing system from natural language input is explained. Our method identifies failures explicitly and implicitly described in natural language by domain experts and uses those descriptions to recommend classifications for inclusion in the experts' heuristic hierarchies.

Author

N93-25980*# Vigyan Research Associates, Inc., Hampton, VA. **MULTI-VIEWPOINT CLUSTERING ANALYSIS**

MALA MEHROTRA and CHRIS WILD (Old Dominion Univ., Norfolk, VA.) In NASA. Goddard Space Flight Center, The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 217-231 1993

(Contract NAS9-18706)

Avail: CASI HC A03/MF A03

In this paper, we address the feasibility of partitioning rule-based systems into a number of meaningful units to enhance the comprehensibility, maintainability and reliability of expert systems software. Preliminary results have shown that no single structuring principle or abstraction hierarchy is sufficient to understand complex knowledge bases. We therefore propose the Multi View Point - Clustering Analysis (MVP-CA) methodology to provide multiple views of the same expert system. We present the results of using this approach to partition a deployed knowledge-based system that navigates the Space Shuttle's entry. We also discuss the impact of this approach on verification and validation of knowledge-based systems.

Author

N93-25981*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

AN APPLICATION OF MACHINE LEARNING TO THE ORGANIZATION OF INSTITUTIONAL SOFTWARE REPOSITORIES

SIDNEY BAILIN (Computer Technology Associates, Inc., Rockville, MD.), SCOTT HENDERSON (Computer Technology Associates, Inc., Rockville, MD.), and WALT TRUSzkowski In its The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 235-248 1993

Avail: CASI HC A03/MF A03

Software reuse has become a major goal in the development of space systems, as a recent NASA-wide workshop on the subject made clear. The Data Systems Technology Division of Goddard Space Flight Center has been working on tools and techniques for promoting reuse, in particular in the development of satellite ground support software. One of these tools is the Experiment in Libraries via Incremental Schemata and Cobweb (ElvisC). ElvisC applies machine learning to the problem of organizing a reusable software component library for efficient and reliable retrieval. In this paper we describe the background factors that have motivated this work, present the design of the system, and evaluate the results of its application.

Author

N93-25982* Analytic Sciences Corp., Reading, MA.
VISUALIZING THE SEMANTIC CONTENT OF LARGE TEXT DATABASES USING TEXT MAPS
NATHAN COMBS In NASA. Goddard Space Flight Center, The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 249-261 1993
Avail: CASI HC A03/MF A03

A methodology for generating text map representations of the semantic content of text databases is presented. Text maps provide a graphical metaphor for conceptualizing and visualizing the contents and data interrelationships of large text databases. Described are a set of experiments conducted against the TIPSTER corpora of Wall Street Journal articles. These experiments provide an introduction to current work in the representation and visualization of documents by way of their semantic content.

Author

N93-25983* Loyola Univ., New Orleans, LA.
IN SEARCH OF META-KNOWLEDGE

ANTONIO M. LOPEZ, JR. In NASA. Goddard Space Flight Center, The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 263-269 1993
Avail: CASI HC A02/MF A03

Development of an Intelligent Information System (IIS) involves application of numerous artificial intelligence (AI) paradigms and advanced technologies. The National Aeronautics and Space Administration (NASA) is interested in an IIS that can automatically collect, classify, store and retrieve data, as well as develop, manipulate and restructure knowledge regarding the data and its application (Campbell et al., 1987, p.3). This interest stems in part from a NASA initiative in support of the interagency Global Change Research program. NASA's space data problems are so large and varied that scientific researchers will find it almost impossible to access the most suitable information from a software system if meta-information (metadata and meta-knowledge) is not embedded in that system. Even if more, faster, larger hardware is used, new innovative software systems will be required to organize, link, maintain, and properly archive the Earth Observing System (EOS) data that is to be stored and distributed by the EOS Data and Information System (EOSDIS) (Dozier, 1990). Although efforts are being made to specify the metadata that will be used in EOSDIS, meta-knowledge specification issues are not clear. With the expectation that EOSDIS might evolve into an IIS, this paper presents certain ideas on the concept of meta-knowledge and demonstrates how meta-knowledge might be represented in a pixel classification problem.

Author

N93-25984* Johns Hopkins Univ., Laurel, MD. Applied Physics Lab.

THE STARVIEW INTELLIGENT QUERY MECHANISM
R. D. SEMMEL and D. P. SILBERBERG (Space Telescope Science Inst., Baltimore, MD.) In NASA. Goddard Space Flight Center, The 1993 Goddard Conference on Space Applications of Artificial Intelligence p 271-285 1993
Avail: CASI HC A03/MF A03

The StarView interface is being developed to facilitate the retrieval of scientific and engineering data produced by the Hubble Space Telescope. While predefined screens in the interface can be used to specify many common requests, ad hoc requests require a dynamic query formulation capability. Unfortunately, logical level knowledge is too sparse to support this capability. In particular, essential formulation knowledge is lost when the domain of interest is mapped to a set of database

relation schemas. Thus, a system known as QUICK has been developed that uses conceptual design knowledge to facilitate query formulation. By heuristically determining strongly associated objects at the conceptual level, QUICK is able to formulate semantically reasonable queries in response to high-level requests that specify only attributes of interest. Moreover, by exploiting constraint knowledge in the conceptual design, QUICK assures that queries are formulated quickly and will execute efficiently.

Author

N94-35043* National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

THE 1994 GODDARD CONFERENCE ON SPACE APPLICATIONS OF ARTIFICIAL INTELLIGENCE
CARL F. HOSTETTER, ed. Washington May 1994 359 p
Conference held in Greenbelt, MD, 10-12 May 1994
(Contract RTOP 510-00-00)

(NASA-CP-3268; REPT-92B00066; NAS 1.55:3268) Avail: CASI HC A16/MF A03

This publication comprises the papers presented at the 1994 Goddard Conference on Space Applications of Artificial Intelligence held at the NASA/GSFC, Greenbelt, Maryland, on 10-12 May 1994. The purpose of this annual conference is to provide a forum in which current research and development directed at space applications of artificial intelligence can be presented and discussed.

N94-35044* International Business Machines Corp., Yorktown Heights, NY.

INTERACTIVE ARCHIVES OF SCIENTIFIC DATA
LLOYD A. TREINISH In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 3-15 May 1994
Avail: CASI HC A03/MF A03

A focus on qualitative methods of presenting data shows that visualization provides a mechanism for browsing independent of the source of data and is an effective alternative to traditional image-based browsing of image data. To be generally applicable, such visualization methods, however, must be based upon an underlying data model with support for a broad class of data types and structures. Interactive, near-real-time browsing for data sets of interesting size today requires a browse server of considerable power. A symmetric multi-processor with very high internal and external bandwidth demonstrates the feasibility of this concept. Although this technology is likely to be available on the desktop within a few years, the increase in the size and complexity of achieved data will continue to exceed the capacity of 'workstation' systems. Hence, a higher class of performance, especially in bandwidth, will generally be required for on-demand browsing. A few experiments with differing digital compression techniques indicates that a MPEG-1 implementation within the context of a high-performance browse server (i.e., parallelized) is a practical method of converting a browse product to a form suitable for network or CD-ROM distribution.

Author

N94-35045* National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

PLANNING APPLICATIONS IN IMAGE ANALYSIS

MARK BODDY (Honeywell, Inc., Minneapolis, MN.), JIM WHITE (Honeywell, Inc., Minneapolis, MN.), ROBERT GOLDMAN (Honeywell, Inc., Minneapolis, MN.), and NICK SHORT, JR. In its The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 17-28 May 1994

Avail: CASI HC A03/MF A03

We describe two interim results from an ongoing effort to automate the acquisition, analysis, archiving, and distribution of satellite earth science data. Both results are applications of Artificial Intelligence planning research to the automatic generation of processing steps for image analysis tasks. First, we have constructed a linear conditional planner (CPed), used to generate conditional processing plans. Second, we have extended an existing hierarchical planning system to make use of durations, resources, and deadlines, thus supporting the automatic generation of processing steps in time and resource-constrained environments.

Author

N94-35046*# Lockheed Missiles and Space Co., Palo Alto, CA.

RAVE: RAPID VISUALIZATION ENVIRONMENT

D. M. KLUMPAR, KEVIN ANDERSON, and AVANGELOS SIMOUDIS In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 29-38 May 1994

Avail: CASI HC A02/MF A03

Visualization is used in the process of analyzing large, multidimensional data sets. However, the selection and creation of visualizations that are appropriate for the characteristics of a particular data set and the satisfaction of the analyst's goals is difficult. The process consists of three tasks that are performed iteratively: generate, test, and refine. The performance of these tasks requires the utilization of several types of domain knowledge that data analysts do not often have. Existing visualization systems and frameworks do not adequately support the performance of these tasks. In this paper we present the RApid Visualization Environment (RAVE), a knowledge-based system that interfaces with commercial visualization frameworks and assists a data analyst in quickly and easily generating, testing, and refining visualizations. RAVE was used for the visualization of in situ measurement data captured by spacecraft.

Author (revised)

N94-35047*# Commonwealth Scientific and Industrial Research Organization, North Ryde (Australia). Div. of Information Technology.

ON-BOARD EMERGENT SCHEDULING OF AUTONOMOUS SPACECRAFT PAYLOAD OPERATIONS

CRAIG A. LINDLEY In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 41-55 May 1994

Avail: CASI HC A03/MF A03

This paper describes a behavioral competency level concerned with emergent scheduling of spacecraft payload operations. The level is part of a multi-level subsumption architecture model for autonomous spacecraft, and it functions as an action selection system for processing a spacecraft commands that can be considered as 'plans-as-communication'. Several versions of the selection mechanism are described, and their robustness is qualitatively compared.

Author (revised)

N94-35048*# Space Telescope Science Inst., Baltimore, MD. **A CRITERION AUTOSCHEDULER FOR LONG RANGE PLANNING**

JEFFREY L. SPONSLER In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 57-71 May 1994

Avail: CASI HC A03/MF A03

A constraint-based scheduling system called SPIKE is used to create long-term schedules for the Hubble Space Telescope. A meta-level scheduler called the Criterion Autoscheduler for Long range planning (CASL) was created to guide SPIKE's schedule generation according to the agenda of the planning scientists. It is proposed that sufficient flexibility exists in a schedule to allow high level planning heuristics to be applied without adversely affected crucial constraints such as spacecraft efficiency. This hypothesis is supported by test data which is described.

Author (revised)

N94-35049*# Space Telescope Science Inst., Baltimore, MD. **PROPAGATING ORIENTATION CONSTRAINTS FOR THE HUBBLE SPACE TELESCOPE**

ASHIM BOSE and ANDY GERB In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 73-86 May 1994

Avail: CASI HC A03/MF A03

An observing program on the Hubble Space Telescope (HST) is described in terms of exposures that are obtained by one or more of the instruments onboard the HST. Many requested exposures might specify orientation requirements and accompanying ranges. Orientation refers to the amount of roll (in degrees) about the line of sight. The range give the permissible tolerance (also in degrees). These requirements may be (1) absolute (in relation to the celestial coordinate system), (2) relative to the nominal roll angle for HST during that exposure, or (3) relative (in relation to other exposures in the observing program). The TRANSformation expert system converts proposals for astronomical observations with HST into detailed observing plans. Part of the conversion involves grouping exposures into higher level structures based on exposure characteristics. Exposures constrained to be at different orientations cannot be grouped together. Because relative orientation requirements cause implicit constraints, orientation constraints have to be propagated. TRANS must also identify any inconsistencies that may exist so they can be corrected. We have designed and implemented an orientation constraint propagator as part of TRANS. The propagator is based on an informal algebra that facilitates the setting up and propagation of the orientation constraints. The constraint propagator generates constraints between directly related exposures, and propagates derived constraints between exposures that are related indirectly. It provides facilities for path-consistency checking, identification of unsatisfiable constraints, and querying of orientation relationships. The system has been successfully operational as part of TRANS for over seven months. The solution has particular significance to space applications in which satellite/telescope pointing and attitude are constrained and relationships exist between multiple configurations.

Author (revised)

N94-35050*# Colorado Univ., Denver, CO. Dept. of Computer Science and Engineering.

A LINGUISTIC GEOMETRY FOR SPACE APPLICATIONS

BORIS STILMAN In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 87-101 May 1994

Avail: CASI HC A03/MF A03

We develop a formal theory, the so-called Linguistic Geometry, in order to discover the inner properties of human expert heuristics, which were successful in a certain class of complex control systems, and apply them to different systems. This research relies on the formalization of search heuristics of

high-skilled human experts which allow for the decomposition of complex system into the hierarchy of subsystems, and thus solve intractable problems reducing the search. The hierarchy of subsystems is represented as a hierarchy of formal attribute languages. This paper includes a formal survey of the Linguistic Geometry, and new example of a solution of optimization problem for the space robotic vehicles. This example includes actual generation of the hierarchy of languages, some details of trajectory generation and demonstrates the drastic reduction of search in comparison with conventional search algorithms.

Author

N94-35051*# Drexel Univ., Philadelphia, PA. IMPAQTC Center.

MULTIRESOLUTIONAL SCHEMATA FOR UNSUPERVISED LEARNING OF AUTONOMOUS ROBOTS FOR 3D SPACE OPERATION

ALBERTO LACAZE, MICHAEL MEYSTERL, and ALEX MEYSTERL In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 103-112 May 1994

Avail: CASI HC A02/MF A03

This paper describes a novel approach to the development of a learning control system for autonomous space robot (ASR) which presents the ASR as a 'baby' -- that is, a system with no a priori knowledge of the world in which it operates, but with behavior acquisition techniques that allows it to build this knowledge from the experiences of actions within a particular environment (we will call it an Astro-baby). The learning techniques are rooted in the recursive algorithm for inductive generation of nested schemata molded from processes of early cognitive development in humans. The algorithm extracts data from the environment and by means of correlation and abduction, it creates schemata that are used for control. This system is robust enough to deal with a constantly changing environment because such changes provoke the creation of new schemata by generalizing from experiences, while still maintaining minimal computational complexity, thanks to the system's multiresolutional nature.

Author

N94-35052*# Martin Marietta Corp., Moorestown, NJ. **A GENETIC TECHNIQUE FOR PLANNING A CONTROL SEQUENCE TO NAVIGATE THE STATE SPACE WITH A QUASI-MINIMUM-COST OUTPUT TRAJECTORY FOR A NON-LINEAR MULTI-DIMENSIONAL SYSTEM** C. HEIN and A. MEYSTERL In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 113-120 May 1994

Avail: CASI HC A02/MF A03

There are many multi-stage optimization problems that are not easily solved through any known direct method when the stages are coupled. For instance, we have investigated the problem of planning a vehicle's control sequence to negotiate obstacles and reach a goal in minimum time. The vehicle has a known mass, and the controlling forces have finite limits. We have developed a technique that finds admissible control trajectories which tend to minimize the vehicle's transit time through the obstacle field. The immediate applications is that of a space robot which must rapidly traverse around 2-or-3 dimensional structures via application of a rotating thruster or non-rotating on-off for such vehicles is located at the Marshall Space Flight Center in Huntsville Alabama. However, it appears that the development method is applicable to a general set of optimization problems in which the cost function and the multi-

dimensional multi-state system can be any nonlinear functions, which are continuous in the operating regions. Other applications included the planning of optimal navigation pathways through a transversability graph; the planning of control input for under-water maneuvering vehicles which have complex control state-space relationships; the planning of control sequences for milling and manufacturing robots; the planning of control and trajectories for automated delivery vehicles; and the optimization and athletic training in slalom sports.

Author

N94-35053*# Old Dominion Univ., Norfolk, VA.

MODELING HETEROGENEOUS PROCESSOR SCHEDULING FOR REAL TIME SYSTEMS

J. F. LEATHRUM, R. R. MIELKE, and J. W. STOUGHTON In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 121-133 May 1994

Avail: CASI HC A03/MF A03

A new model is presented to describe dataflow algorithms implemented in a multiprocessor system. Called the resource/data flow graph (RDFG), the model explicitly represents cyclo-static processor schedules as circuits of processor arcs which reflect the order that processors execute graph nodes. The model also allows the guarantee of meeting hard real-time deadlines. When unfolded, the model identifies statically the processor schedule. The model therefore is useful for determining the throughput and latency of systems with heterogeneous processors. The applicability of the model is demonstrated using a space surveillance algorithm.

Author

N94-35054*# Allied-Signal Aerospace Co., Seabrook, MD. **C++ PLANNING AND RESOURCE REASONING (PARR) SHELL**

JAMES MCINTYRE, ALAN TUCHMAN, DAVID MCLEAN, and RONALD LITTLEFIELD In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 135-146 May 1994

(Contract NASS-27772)

Avail: CASI HC A03/MF A03

This paper describes a generic, C++ version of the Planning and Resource Reasoning (PARR) shell which has been developed to supersede the C-based versions of PARR that are currently used to support AI planning and scheduling applications in flight operations centers at Goddard Space Flight Center. This new object-oriented version of PARR can be more easily customized to build a variety of planning and scheduling applications, and C++ PARR applications can be more easily ported to different environments. Genetic classes, constraints, strategies, and paradigms are described along with two types of PARR interfaces.

Author

N94-35055*# Pennsylvania State Univ., University Park, PA. Dept. of Computer Science and Engineering.

ACCURATE ESTIMATION OF OBJECT LOCATION IN AN IMAGE SEQUENCE USING HELICOPTER FLIGHT DATA

YUAN-LIANG TANG and RANGACHAR KASTURI In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 147-157 May 1994

(Contract NAG1-1371)

Avail: CASI HC A03/MF A03

In autonomous navigation, it is essential to obtain a three-dimensional (3D) description of the static environment in

which the vehicle is traveling. For a rotorcraft conducting low-latitude flight, this description is particularly useful for obstacle detection and avoidance. In this paper, we address the problem of 3D position estimation for static objects from a monocular sequence of images captured from a low-latitude flying helicopter. Since the environment is static, it is well known that the optical flow in the image will produce a radiating pattern from the focus of expansion. We propose a motion analysis system which utilizes the epipolar constraint to accurately estimate 3D positions of scene objects in a real world image sequence taken from a low-altitude flying helicopter. Results show that this approach gives good estimates of object positions near the rotorcraft's intended flight-path.

Author (revised)

N94-35056*# Massachusetts Univ., Amherst, MA. Computer Science Dept.

AUTOMATED ANALYSIS OF COMPLEX DATA

ROBERT SAINTAMANT and PAUL R. COHEN In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 161-171 May 1994

(Contract F30602-91-C-0076; F30602-93-C-0100)

Avail: CASI HC A03/MF A03

We have examined some of the issues involved in automating exploratory data analysis, in particular the tradeoff between control and opportunism. We have proposed an opportunistic planning solution for this tradeoff, and we have implemented a prototype, Igor, to test the approach. Our experience in developing Igor was surprisingly smooth. In contrast to earlier versions that relied on rule representation, it was straightforward to increment Igor's knowledge base without causing the search space to explode. The planning representation appears to be both general and powerful, with high level strategic knowledge provided by goals and plans, and the hooks for domain-specific knowledge are provided by monitors and focusing heuristics.

Author (revised)

N94-35057*# Caelum Research Corp., Silver Spring, MD.

AN ADAPTIVE TECHNIQUE TO MAXIMIZE LOSSLESS IMAGE DATA COMPRESSION OF SATELLITE IMAGES

ROBERT J. STEWART, Y. M. FLEMING LURE, and C. S. JOE LIOU In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 173-182 May 1994

Avail: CASI HC A02/MF A03

Data compression will play an increasingly important role in the storage and transmission of image data within NASA science programs as the Earth Observing System comes into operation. It is important that the science data be preserved at the fidelity the instrument and the satellite communication systems were designed to produce. Lossless compression must therefore be applied, at least, to archive the processed instrument data. In this paper, we present an analysis of the performance of lossless compression techniques and develop an adaptive approach which applied image remapping, feature-based image segmentation to determine regions of similar entropy and high-order arithmetic coding to obtain significant improvements over the use of conventional compression techniques alone. Image remapping is used to transform the original image into a lower entropy state. Several techniques were tested on satellite images including differential pulse code modulation, bi-linear interpolation, and block-based linear predictive coding. The results of these experiments are discussed and trade-offs between computation requirements and entropy reductions are used to identify the

optimum approach for a variety of satellite images. Further entropy reduction can be achieved by segmenting the image based on local entropy properties then applying a coding technique which maximizes compression for the region. Experimental results are presented showing the effect of different coding techniques for regions of different entropy. A rule-base is developed through which the technique giving the best compression is selected. The paper concludes that maximum compression can be achieved cost effectively and at acceptable performance rates with a combination of techniques which are selected based on image contextual information.

Author (revised)

N94-35058*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

AUTOMATIC CATALOGUING AND CHARACTERIZATION OF EARTH SCIENCE DATA USING SE-TREES

RON RYMON (Pennsylvania Univ., Philadelphia.) and NICHOLAS M. SHORT, JR. In its The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 183-192 May 1994

(Contract DAAL03-89-C-0031)

Avail: CASI HC A02/MF A03

In the future, NASA's Earth Observing System (EOS) platforms will produce enormous amounts of remote sensing image data that will be stored in the EOS Data Information System. For the past several years, the Intelligent Data Management group at Goddard's Information Science and Technology Office has been researching techniques for automatically cataloguing and characterizing image data (ADCC) from EOS into a distributed database. At the core of the approach, scientists will be able to retrieve data based upon the contents of the imagery. The ability to automatically classify imagery is key to the success of contents-based search. We report results from experiments applying a novel machine learning framework, based on Set-Enumeration (SE) trees, to the ADCC domain. We experiment with two images: one taken from the Blackhills region in South Dakota; and the other from the Washington DC area. In a classical machine learning experimentation approach, an image's pixels are randomly partitioned into training (i.e. including ground truth or survey data) and testing sets. The prediction model is built using the pixels in the training set, and its performance is estimated using the testing set. With the first Blackhills image, we perform various experiments achieving an accuracy level of 83.2 percent, compared to 72.7 percent using a Back Propagation Neural Network (BPNN) and 65.3 percent using a Gaussian Maximum Likelihood Classifier (GMLC). However, with the Washington DC image, we were only able to achieve 71.4 percent, compared with 67.7 percent reported for the BPNN model and 62.3 percent for the GMLC.

Author (revised)

N94-35059*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

VEG: AN INTELLIGENT WORKBENCH FOR ANALYSING SPECTRAL REFLECTANCE DATA

P. ANN HARRISON (JJM Systems, Inc., Arlington, VA.), PATRICK R. HARRISON (Naval Academy, Annapolis, MD.), and DANIEL S. KIMES In its The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 193-205 May 1994

Avail: CASI HC A03/MF A03

An Intelligent Workbench (VEG) was developed for

the systematic study of remotely sensed optical data from vegetation. A goal of the remote sensing community is to infer the physical and biological properties of vegetation cover (e.g. cover type, hemispherical reflectance, ground cover, leaf area index, biomass, and photosynthetic capacity) using directional spectral data. VEG collects together, in a common format, techniques previously available from many different sources in a variety of formats. The decision as to when a particular technique should be applied is nonalgorithmic and requires expert knowledge. VEG has codified this expert knowledge into a rule-based decision component for determining which technique to use. VEG provides a comprehensive interface that makes applying the techniques simple and aids a researcher in developing and testing new techniques. VEG also provides a classification algorithm that can learn new classes of surface features. The learning system uses the database of historical cover types to learn class descriptions of one or more classes of cover types.

Author (revised)

N94-35060*# National Aeronautics and Space Administration. Goddard Space Flight Center, Greenbelt, MD.

A STATISTICAL INFERENCE APPROACH FOR THE RETRIEVAL OF THE ATMOSPHERIC OZONE PROFILE FROM SIMULATED SATELLITE MEASUREMENTS OF SOLAR BACKSCATTERED ULTRAVIOLET RADIATION

N. L. BONAVITO, C. L. GORDON, R. INGUVA, G. N. SERAFINO, and R. A. BARNES In its The 1994 Goddard Conference on Space Applications of Artificial Intelligence 207-222 May 1994

Avail: CASI HC A03/MF A03

NASA's Mission to Planet Earth (MTPE) will address important interdisciplinary and environmental issues such as global warming, ozone depletion, deforestation, acid rain, and the like with its long term satellite observations of the Earth and with its comprehensive Data and Information System. Extensive sets of satellite observations supporting MTPE will be provided by the Earth Observing System (EOS), while more specific process related observations will be provided by smaller Earth Probes. MTPE will use data from ground and airborne scientific investigations to supplement and validate the global observations obtained from satellite imagery, while the EOS satellites will support interdisciplinary research and model development. This is important for understanding the processes that control the global environment and for improving the prediction of events. In this paper we illustrate the potential for powerful artificial intelligence (AI) techniques when used in the analysis of the formidable problems that exist in the NASA Earth Science programs and of those to be encountered in the future MTPE and EOS programs. These techniques, based on the logical and probabilistic reasoning aspects of plausible inference, strongly emphasize the synergetic relation between data and information. As such, they are ideally suited for the analysis of the massive data streams to be provided by both MTPE and EOS. To demonstrate this, we address both the satellite imagery and model enhancement issues for the problem of ozone profile retrieval through a method based on plausible scientific inferencing. Since in the retrieval problem, the atmospheric ozone profile that is consistent with a given set of measured radiances may not be unique, an optimum statistical method is used to estimate a 'best' profile solution from the radiances and from additional a priori information.

Author

N94-35061*#

Indian Space Research Organization, Ahmedabad.

APPLICATION OF ARTIFICIAL NEURAL NETWORKS IN HYDROLOGICAL MODELING: A CASE STUDY OF RUNOFF SIMULATION OF A HIMALAYAN GLACIER BASIN

A. M. BUCH, A. NARAIN, and P. C. PANDEY In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 223-230 May 1994

Avail: CASI HC A02/MF A03

The simulation of runoff from a Himalayan Glacier basin using an Artificial Neural Network (ANN) is presented. The performance of the ANN model is found to be superior to the Energy Balance Model and the Multiple Regression model. The RMS Error is used as the figure of merit for judging the performance of the three models, and the RMS Error for the ANN model is the latest of the three models. The ANN is faster in learning and exhibits excellent system generalization characteristics.

Author

N94-35062*# Illinois Univ., Urbana-Champaign, IL. Dept. of Mechanical and Industrial Engineering.

INTELLIGENT RESOURCES FOR SATELLITE GROUND CONTROL OPERATIONS

PATRICIA M. JONES In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 233-239 May 1994 (Contract NAG5-244; NSF IRI-92-10918)

Avail: CASI HC A02/MF A03

This paper describes a cooperative approach to the design of intelligent automation and describes the Mission Operations Cooperative Assistant for NASA Goddard flight operations. The cooperative problem solving approach is being explored currently in the context of providing support for human operator teams and also in the definition of future advanced automation in ground control systems.

Author

N94-35063*# National Aeronautics and Space Administration. Lyndon B. Johnson Space Center, Houston, TX.

VISTA GOES ONLINE: DECISION-ANALYTIC SYSTEMS FOR REAL-TIME DECISION-MAKING IN MISSION CONTROL

MATTHEW BARRY, ERIC HORVITZ, CORINNE RUOKANGAS, and SAMPATH SRINIVAS In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 241-252 May 1994

Avail: CASI HC A03/MF A03

The Vista project has centered on the use of decision-theoretic approaches for managing the display of critical information relevant to real-time operations decisions. The Vista-I project originally developed a prototype of these approaches for managing flight control displays in the Space Shuttle Mission Control Center (MCC). The follow-on Vista-II project integrated these approaches in a workstation program which currently is being certified for use in the MCC. To our knowledge, this will be the first application of automated decision-theoretic reasoning techniques for real-time spacecraft operations. We shall describe the development and capabilities of the Vista-II system, and provide an overview of the use of decision-theoretic reasoning techniques to the problems of managing the complexity of flight controller displays. We discuss the relevance of the Vista techniques within the MCC decision-making environment, focusing on the problems of detecting and diagnosing spacecraft

electromechanical subsystems component failures with limited information, and the problem of determining what control actions should be taken in high-stakes, time-critical situations in response to a diagnosis performed under uncertainty. Finally, we shall outline our current research directions for follow-on projects.

Author

N94-35064*# National Aeronautics and Space Administration.
Lyndon B. Johnson Space Center, Houston, TX.

**MISSION EVALUATION ROOM INTELLIGENT
DIAGNOSTIC AND ANALYSIS SYSTEM (MIDAS)**

GINGER L. PACK, JANE FALGOUT, JOSEPH BARCIO,
STEVE SHNURER, DAVID WADSWORTH, and LOUIS
FLORES In NASA. Goddard Space Flight Center, The 1994
Goddard Conference on Space Applications of Artificial
Intelligence p 253-267 May 1994

Avail: CASI HC A03/MF A03

The role of Mission Evaluation Room (MER) engineers is to provide engineering support during Space Shuttle missions, for Space Shuttle systems. These engineers are concerned with ensuring that the systems for which they are responsible function reliably, and as intended. The MER is a central facility from which engineers may work, in fulfilling this obligation. Engineers participate in real-time monitoring of shuttle telemetry data and provide a variety of analyses associated with the operation of the shuttle. The Johnson Space Center's Automation and Robotics Division is working to transfer advances in intelligent systems technology to NASA's operational environment. Specifically, the MER Intelligent Diagnostic and Analysis System (MIDAS) project provides MER engineers with software to assist them with monitoring, filtering and analyzing Shuttle telemetry data, during and after Shuttle missions. MIDAS off-loads to computers and software, the tasks of data gathering, filtering, and analysis, and provides the engineers with information which is in a more concise and usable form needed to support decision making and engineering evaluation. Engineers are then able to concentrate on more difficult problems as they arise. This paper describes some, but not all of the applications that have been developed for MER engineers, under the MIDAS Project. The sampling described herewith was selected to show the range of tasks that engineers must perform for mission support, and to show the various levels of automation that have been applied to assist their efforts.

Author

N94-35065*# State Univ. of New York, Binghamton, NY.
**QUALITATIVE MODEL-BASED DIAGNOSIS USING
POSSIBILITY THEORY**

CLIFF JOSLYN In NASA. Goddard Space Flight Center, The
1994 Goddard Conference on Space Applications of Artificial
Intelligence p 269-283 May 1994

(Contract NGT-50756)

Avail: CASI HC A03/MF A03

The potential for the use of possibility in the qualitative model-based diagnosis of spacecraft systems is described. The first sections of the paper briefly introduce the Model-Based Diagnostic (MBD) approach to spacecraft fault diagnosis; Qualitative Modeling (QM) methodologies; and the concepts of possibilistic modeling in the context of Generalized Information Theory (GIT). Then the necessary conditions for the applicability of possibilistic methods to qualitative MBD, and a number of potential directions for such an application, are described.

Author

N94-35066*# Temple Univ., Philadelphia, PA. Dept. of
Electrical Engineering.

**AN INTELLIGENT CONTROL SYSTEM FOR FAILURE
DETECTION AND CONTROLLER RECONFIGURATION**
SAROJ K. BISWAS In NASA. Goddard Space Flight Center,
The 1994 Goddard Conference on Space Applications of
Artificial Intelligence p 285-291 May 1994

Avail: CASI HC A02/MF A03

We present an architecture of an intelligent restructurable control system to automatically detect failure of system components, assess its impact on system performance and safety, and reconfigure the controller for performance recovery. Fault detection is based on neural network associative memories and pattern classifiers, and is implemented using a multilayer feedforward network. Details of the fault detection network along with simulation results on health monitoring of a dc motor have been presented. Conceptual developments for fault assessment using an expert system and controller reconfiguration using a neural network are outlined.

Author

N94-35067*# University of Southern California, Marina del
Rey, CA. Information Sciences Inst.

**TELECOMMUNICATIONS ISSUES OF INTELLIGENT
DATABASE MANAGEMENT FOR GROUND
PROCESSING SYSTEMS IN THE EOS ERA**

JOSEPH D. TOUCH In NASA. Goddard Space Flight Center,
The 1994 Goddard Conference on Space Applications of
Artificial Intelligence p 295-303 May 1994

(Contract DABT63-91-C-0001)

Avail: CASI HC A02/MF A03

Future NASA earth science missions, including the Earth Observing System (EOS), will be generating vast amounts of data that must be processed and stored at various locations around the world. Here we present a stepwise-refinement of the intelligent database management (IDM) of the distributed active archive center (DAAC - one of seven regionally-located EOSDIS archive sites) architecture, to showcase the telecommunications issues involved. We develop this architecture into a general overall design. We show that the current evolution of protocols is sufficient to support IDM at Gbps rates over large distances. We also show that network design can accommodate a flexible data ingestion storage pipeline and a user extraction and visualization engine, without interference between the two.

Author

N94-35068*# Symbiotics, Inc., Cambridge, MA.

**GROUP-ORIENTED COORDINATION MODELS FOR
DISTRIBUTED CLIENT-SERVER COMPUTING**

RICHARD M. ADLER and CRAIG S. HUGHES In NASA.
Goddard Space Flight Center, The 1994 Goddard Conference on
Space Applications of Artificial Intelligence p 305-318 May
1994

(Contract NAS8-39343; NAS8-39905)

Avail: CASI HC A03/MF A03

This paper describes group-oriented control models for distributed client-server interactions. These models transparently coordinate requests for services that involve multiple servers, such as queries across distributed databases. Specific capabilities include: decomposing and replicating client requests; dispatching request subtasks or copies to independent, networked servers; and combining server results into a single response for the client. The control models were implemented by combining request broker and process group technologies with an object-oriented communication middleware tool. The models are illustrated in

the context of a distributed operations support application for space-based systems.

Author

N94-35069*# National Aeronautics and Space Administration. Lyndon B. Johnson Space Center, Houston, TX.

A DISTRIBUTED COMPUTING MODEL FOR TELEMETRY DATA PROCESSING

MATTHEW R. BARRY (Rockwell Space Operations Co., Houston, TX.), KEVIN L. SCOTT, and STEVEN P. WEISMULLER In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 319-328 May 1994

Avail: CASI HC A02/MF A03

We present a new approach to distributing processed telemetry data among spacecraft flight controllers within the control centers at NASA's Johnson Space Center. This approach facilitates the development of application programs which integrate spacecraft-telemetered data and ground-based synthesized data, then distributes this information to flight controllers for analysis and decision-making. The new approach combines various distributed computing models into one hybrid distributed computing model. The model employs both client-server and peer-to-peer distributed computing models cooperating to provide users with information throughout a diverse operations environment. Specifically, it provides an attractive foundation upon which we are building critical real-time monitoring and control applications, while simultaneously lending itself to peripheral applications in playback operations, mission preparations, flight controller training, and program development and verification. We have realized the hybrid distributed computing model through an information sharing protocol. We shall describe the motivations that inspired us to create this protocol, along with a brief conceptual description of the distributed computing models it employs. We describe the protocol design in more detail, discussing many of the program design considerations and techniques we have adopted. Finally, we describe how this model is especially suitable for supporting the implementation of distributed expert system applications.

Author

N94-35070*# Aerospace Corp., El Segundo, CA. Computer Systems Div.

ISTAR: INTELLIGENT SYSTEM FOR TELEMETRY ANALYSIS IN REAL-TIME

CHARLES SIMMONS In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 329-342 May 1994

Avail: CASI HC A03/MF A03

The intelligent system for telemetry analysis in real-time (ISTAR) is an advanced vehicle monitoring environment incorporating expert systems, analysis tools, and on-line hypermedia documentation. The system was developed for the Air Force Space and Missile Systems Center (SMC) in Los Angeles, California, in support of the inertial upper stage (IUS) booster vehicle. Over a five year period the system progressed from rapid prototype to operational system. ISTAR has been used to support five IUS missions and countless mission simulations. There were a significant number of lessons learned with respect to integrating an expert system capability into an existing ground system.

Author (revised)

N94-35071*# Durham Univ. (England). Artificial intelligence Systems Research Group.

ENGINEERING LARGE-SCALE AGENT-BASED

SYSTEMS WITH CONSENSUS

A. BOKMA, A. SLADE, S. KERRIDGE, and K. JOHNSON In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 343-356 May 1994

Avail: CASI HC A03/MF A03

The paper presents the consensus method for the development of large-scale agent-based systems. Systems can be developed as networks of knowledge based agents (KBA) which engage in a collaborative problem solving effort. The method provides a comprehensive and integrated approach to the development of this type of system. This includes a systematic analysis of user requirements as well as a structured approach to generating a system design which exhibits the desired functionality. There is a direct correspondence between system requirements and design components. The benefits of this approach are that requirements are traceable into design components and code thus facilitating verification. The use of the consensus method with two major test applications showed it to be successful and also provided valuable insight into problems typically associated with the development of large systems.

Author (revised)

N94-35072*# BSO/Aerospace and Systems B.V., Nieuwegein (Netherlands).

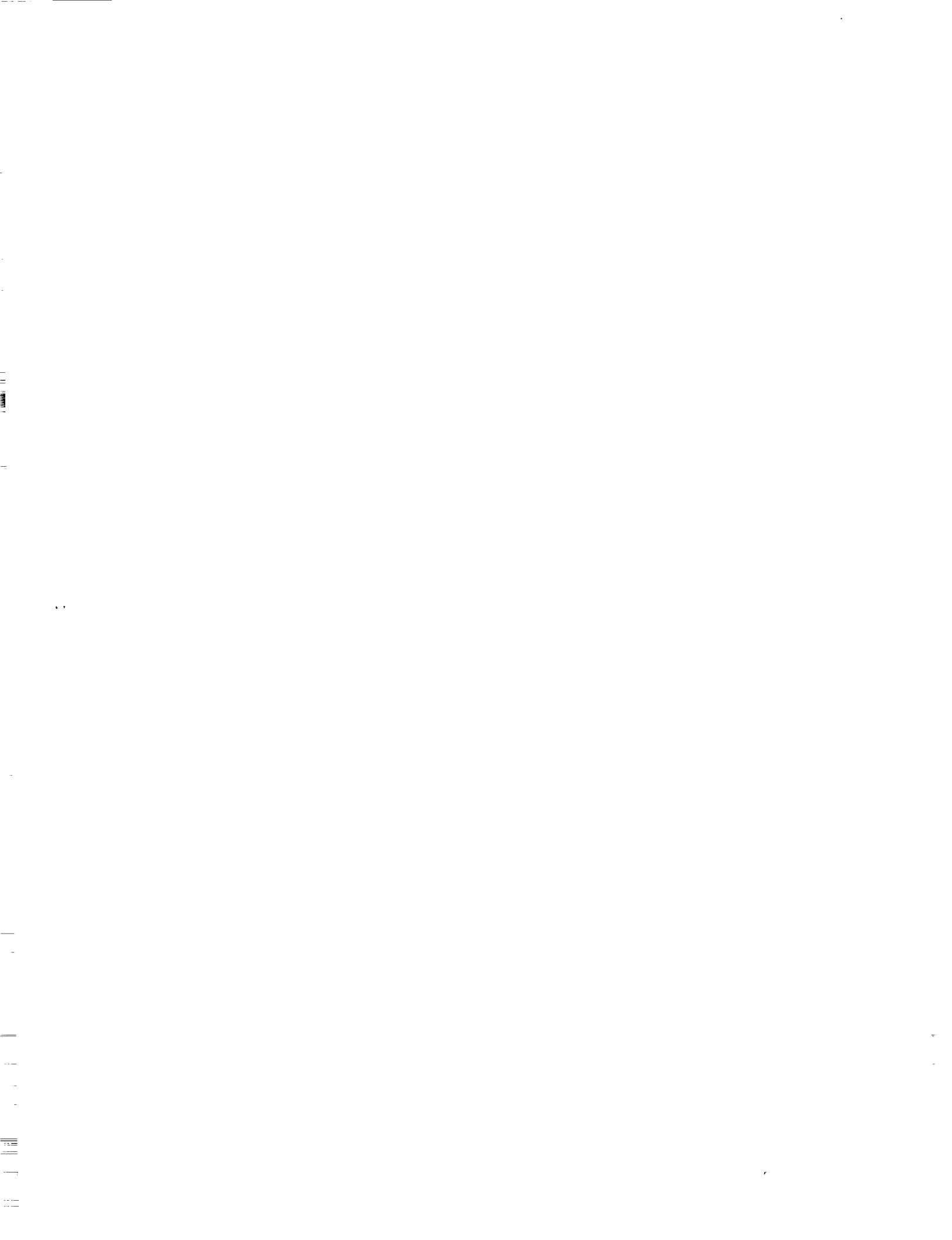
SPACE/GROUND SYSTEMS AS COOPERATING AGENTS

T. J. GRANT In NASA. Goddard Space Flight Center, The 1994 Goddard Conference on Space Applications of Artificial Intelligence p 357-368 May 1994

Avail: CASI HC A03/MF A03

Within NASA and the European Space Agency (ESA) it is agreed that autonomy is an important goal for the design of future spacecraft and that this requires on-board artificial intelligence. NASA emphasizes deep space and planetary rover missions, while ESA considers on-board autonomy as an enabling technology for missions that must cope with imperfect communications. ESA's attention is on the space/ground system. A major issue is the optimal distribution of intelligent functions within the space/ground system. This paper describes the multi-agent architecture for space/ground systems (MAASGS) which would enable this issue to be investigated. A MAASGS agent may model a complete spacecraft, a spacecraft subsystem or payload, a ground segment, a spacecraft control system, a human operator, or an environment. The MAASGS architecture has evolved through a series of prototypes. The paper recommends that the MAASGS architecture should be implemented in the operational Dutch Utilization Center.

Author (revised)



SUBJECT INDEX

A

ACCURACY

Classifying multispectral data by neural networks N93-25976

ACTIVE CONTROL

A hierarchical distributed control model for coordinating intelligent systems N91-22783

ACTS

Intelligent fault isolation and diagnosis for communication satellite systems N92-23365

ACTUATORS

Actuators for a space manipulator N89-10101

Parametric motion control of robotic arms: A biologically based approach using neural networks N93-25964

ADA (PROGRAMMING LANGUAGE)

A LISP-Ada connection N89-10085

Techniques and implementation of the embedded rule-based expert system using Ada N91-22788

ADAPTIVE CONTROL

An adaptive control scheme for a flexible manipulator N89-10095

Computer hardware and software for robotic control N89-10103

Multiresolutional schemata for unsupervised learning of autonomous robots for 3D space operation N94-35051

AEROSPACE ENGINEERING

The 1988 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3009) N88-30330

The 1992 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3141) N92-23356

AEROSPACE SYSTEMS

Representing functions/procedures and processes/structures for analysis of effects of failures on functions and operations N91-22780

The achievement of spacecraft autonomy through the thematic application of multiple cooperating intelligent agents N92-23364

An application of machine learning to the organization of institutional software repositories N93-25981

Multiresolutional schemata for unsupervised learning of autonomous robots for 3D space operation N94-35051

AIRCRAFT ACCIDENT INVESTIGATION

The use of multiple models in case-based diagnosis N93-25969

AIRCRAFT ACCIDENTS

The use of multiple models in case-based diagnosis N93-25969

AIRCRAFT CONTROL

Applications of fuzzy sets to rule-based expert system development N89-26606

ALGORITHMS

Range data description based on multiple characteristics N88-30351

AUTOPLAN: A PC-based automated mission planning tool N89-10067

A heuristic approach to incremental and reactive scheduling N89-26589

Genetic algorithms applied to the scheduling of the Hubble Space Telescope N89-26607

An approach to rescheduling activities based on determination of priority and disruptivity N90-22299

Space communications scheduler: A rule-based approach to adaptive deadline scheduling N90-22300

Bounded-time fault-tolerant rule-based systems N90-22318

AI techniques for a space application scheduling problem N91-22776

Improved interpretation of satellite altimeter data using genetic algorithms N92-23369

Adaptive laser link reconfiguration using constraint propagation N93-25967

ALLOCATIONS

Dynamic task allocation for a man-machine symbiotic system N89-10098

ALTERNATIVES

Qualitative model-based diagnosis using possibility theory N94-35065

ANALOGIES

The use of multiple models in case-based diagnosis N93-25969

ANOMALIES

Expert system support for HST operations N89-10076

A machine independent expert system for diagnosing environmentally induced spacecraft anomalies N91-22782

Anomalous event diagnosis for environmental satellite systems N93-25971

APPLICATIONS PROGRAMS (COMPUTERS)

A heuristic approach to incremental and reactive scheduling N89-26589

A criterion autoscheduler for long range planning N94-35048

ARCHITECTURE (COMPUTERS)

Automated satellite control in Ada N88-30336

MOORE: A prototype expert system for diagnosing spacecraft problems N88-30343

A hierarchically distributed architecture for fault isolation expert systems on the space station N89-10077

NASREN: Standard reference model for telerobot control N89-10099

Teleoperated position control of a PUMA robot N89-10104

Ground data systems resource allocation process N89-26582

A situated reasoning architecture for space-based repair and replace tasks N89-26583

On the development of a reactive sensor-based robotic system N89-26586

The 1990 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3068) N90-22294

Resource Allocation Planning Helper (RALPH): Lessons learned N90-22296

The role of artificial intelligence techniques in scheduling systems N90-22297

SURE (Science User Resource Expert): A science planning and scheduling assistant for a resource based environment N90-22302

Managing temporal relations N90-22304

Autonomous power expert system N90-22306

Spacecraft command verification: The AI solution N90-22307

Adaptive pattern recognition by mini-max neural networks as a part of an intelligent processor N90-22317

Long range science scheduling for the Hubble Space Telescope N91-22775

SLS-PLAN-JT: A knowledge-based blackboard scheduling system for Spacelab life sciences missions N92-23358

An architecture for the development of real-time fault diagnosis systems using model-based reasoning N92-23363

Feature detection in satellite images using neural network technology N92-23366

An intelligent control system for failure detection and controller reconfiguration N94-35066

ARTIFICIAL INTELLIGENCE

The 1988 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3009) N88-30330

An intelligent training system for space shuttle flight controllers N88-30331

Artificial intelligence costs, benefits, risks for selected spacecraft ground system automation scenarios N88-30332

A shared-world conceptual model for integrating space station life sciences telescience operations N88-30333

Artificial intelligence in a mission operations and satellite test environment N88-30334

Automated satellite control in Ada	N88-30336
Contingency rescheduling of spacecraft operations	N88-30337
Knowledge based tools for Hubble Space Telescope planning and scheduling: Constraints and strategies	N88-30338
The proposal entry processor: Telescience applications for Hubble Space Telescope science operations	N88-30339
Candidate functions for advanced technology implementation in the Columbus mission planning environment	N88-30340
A rule-based systems approach to spacecraft communications configuration optimization	N88-30341
MOORE: A prototype expert system for diagnosing spacecraft problems	N88-30343
Low level image processing techniques using the pipeline image processing engine in the flight telerobotic servicer	N88-30346
Autonomous image data reduction by analysis and interpretation	N88-30347
Parallel and distributed computation for fault-tolerant object recognition	N88-30350
The second generation intelligent user interface for the crustal dynamics data information system	N88-30352
Spacelab data processing facility (SLDPF) Quality Assurance (QA)/Data Accounting (DA) expert systems: Transition from prototypes to operational systems	N88-30353
Automated cataloging and characterization of space-derived data	N88-30354
A design for a ground-based data management system	N88-30355
Automatic mathematical modeling for real time simulation system	N88-30356
A simulation engine: Combining an expert system with a simulation language	N88-30358
Verification and validation of rulebased systems for Hubble Space Telescope ground support	N88-30361
The need for a comprehensive expert system development methodology	N88-30362
Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics (NASA-TM-89663)	N89-10063
Maintaining an expert system for the Hubble Space Telescope ground support	N89-10064
PLAN-IT: Scheduling assistant for solar system exploration	N89-10068
Deep space network resource scheduling approach and application	N89-10070
Space station Platform Management System (PMS) replanning using resource envelopes	N89-10071
The load shedding advisor: An example of a crisis-response expert system	N89-10073
FIESTA: An operational decision aid for space network fault isolation	N89-10075
Expert system support for HST operations	N89-10076
Toward an expert project management system	N89-10082
Expert systems built by the Expert: An evaluation of OPS5	N89-10086
Issues, concerns, and initial implementation results for space based telerobotic control	N89-10091
Computer hardware and software for robotic control	N89-10103
The 1989 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3033)	N89-26578
Shared resource control between human and computer	N89-26580
An English language interface for constrained domains	N89-26581
A situated reasoning architecture for space-based repair and replace tasks	N89-26583
Parallel plan execution with self-processing networks	N89-26584
Mission scheduling	N89-26585
On the development of a reactive sensor-based robotic system	N89-26586
PST and PARR: Plan specification tools and a planning and	
resource reasoning shell for use in satellite mission planning	N89-26587
A heuristic approach to incremental and reactive scheduling	N89-26589
SHARP: A multi-mission AI system for spacecraft telemetry monitoring and diagnosis	N89-26594
Diagnostic tolerance for missing sensor data	N89-26596
A rapid prototyping/artificial intelligence approach to space station-era information management and access	N89-26600
The 1990 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3068)	N90-22294
The application of connectionism to query planning/scheduling in intelligent user interfaces	N90-22295
Resource Allocation Planning Helper (RALPH): Lessons learned	N90-22296
The role of artificial intelligence techniques in scheduling systems	N90-22297
PLAN-IT-2: The next generation planning and scheduling tool	N90-22298
Space communications scheduler: A rule-based approach to adaptive deadline scheduling	N90-22300
Constraint-based evaluation of sequential procedures	N90-22301
Managing temporal relations	N90-22304
Spacecraft command verification: The AI solution	N90-22307
Analyzing spacecraft configurations through specialization and default reasoning	N90-22308
Simulation-based reasoning about the physical propagation of fault effects	N90-22309
The JPL/KSC telerobotic inspection demonstration	N90-22312
System control of an autonomous planetary mobile spacecraft	N90-22313
Automated procedure execution for space vehicle autonomous control	N90-22314
Knowledge structure representation and automated updates in intelligent information management systems	N90-22316
Bounded-time fault-tolerant rule-based systems	N90-22318
A knowledge-based system with learning for computer communication network design	N90-22319
The 1991 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3110)	N91-22769
Sharing intelligence: Decision-making interactions between users and software in MAESTRO	N91-22772
Using C to build a satellite scheduling expert system: Examples from the Explorer platform planning system	N91-22774
Long range science scheduling for the Hubble Space Telescope	N91-22775
AI techniques for a space application scheduling problem	N91-22776
A failure diagnosis and impact assessment prototype for Space Station Freedom	N91-22777
A failure recovery planning prototype for Space Station Freedom	N91-22778
The Generic Spacecraft Analyst Assistant (GenSAA): A tool for automating spacecraft monitoring with expert systems	N91-22779
Representing functions/procedures and processes/structures for analysis of effects of failures on functions and operations	N91-22780
Autonomous power system intelligent diagnosis and control	N91-22781
A machine independent expert system for diagnosing environmentally induced spacecraft anomalies	N91-22782
A hierarchical distributed control model for coordinating intelligent systems	N91-22783
A color-coded vision scheme for robotics	N91-22785
Expert operator's associate: A knowledge based system for spacecraft control	N91-22786
Techniques and implementation of the embedded rule-based expert system using Ada	N91-22788
A reusable knowledge acquisition shell: KASH	N91-22789
Capturing flight system test engineering expertise: Lessons learned	N91-22790

A proven knowledge-based approach to prioritizing process information	N91-22791
Design of an intelligent information system for in-flight emergency assistance	N91-22792
Preliminary results of investigations into the use of artificial neural networks for discriminating gas chromatograph mass spectra of remote samples	N91-22793
Metamorphoses of ONAV console operations: From prototype to real time application	N91-22794
Lessons learned in the development of the STOL intelligent tutoring system	N91-22795
Space Communications Artificial Intelligence for Link Evaluation Terminal (SCAILET)	N91-22796
Knowledge repositories for multiple uses	N91-22797
The 1992 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3141)	N92-23356
Artificial intelligence approach to planning the robotic assembly of large tetrahedral truss structures	N92-23357
Detecting opportunities for parallel observations on the Hubble Space Telescope	N92-23359
Coordinating complex problem-solving among distributed intelligent agents	N92-23360
An architecture for the development of real-time fault diagnosis systems using model-based reasoning	N92-23363
Improved image classification with neural networks by fusing multispectral signatures with topological data	N92-23368
The use of artificial intelligence techniques to improve the multiple payload integration process	N92-23370
Logic programming and metadata specifications	N92-23375
The 1993 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3200)	N93-25961
Using AI/expert system technology to automate planning and replanning for the HST servicing missions	N93-25962
Adaptive laser link reconfiguration using constraint propagation	N93-25967
An architecture for object-oriented intelligent control of power systems in space	N93-25968
An autonomous satellite architecture integrating deliberative reasoning and behavioural intelligence	N93-25970
Visualizing the semantic content of large text databases using text maps	N93-25982
The StarView intelligent query mechanism	N93-25984
The 1994 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3268)	N94-35043
Planning applications in image analysis	N94-35045
On-board emergent scheduling of autonomous spacecraft payload operations	N94-35047
C++ Planning and Resource Reasoning (PARR) shell	N94-35054
Automated analysis of complex data	N94-35056
A statistical inference approach for the retrieval of the atmospheric ozone profile from simulated satellite measurements of solar backscattered ultraviolet radiation	N94-35060
Space/ground systems as cooperating agents	N94-35072
Proceedings of 1986 Conference on Artificial Intelligence Applications (NASA-TM-101873)	N90-70235
The development of an intelligent user interface for NASA's scientific databases	N90-70236
AI goes Forth	N90-70237
An expert system for ground support of the Hubble space telescope	N90-70239
Simple methods of exploiting the underlying structure of rule-based systems	N90-70244
ARTIFICIAL SATELLITES	
On-board emergent scheduling of autonomous spacecraft payload operations	N94-35047
ASSEMBLING	
Optimum-AIV: A planning and scheduling system for spacecraft AIV	N91-22770
ASTRONOMY	
An expert system approach to astronomical data analysis	N89-10080
Expert systems built by the Expert: An evaluation of OPS5	N89-10086
ATMOSPHERIC COMPOSITION	
A statistical inference approach for the retrieval of the atmospheric ozone profile from simulated satellite measurements of solar backscattered ultraviolet radiation	N94-35060
ATTITUDE CONTROL	
MOORE: A prototype expert system for diagnosing spacecraft problems	N88-30343
Tracking and data relay satellite fault isolation and correction using PACES: Power and attitude control expert system	N89-26592
AUTOMATIC CONTROL	
Artificial intelligence costs, benefits, risks for selected spacecraft ground system automation scenarios	N88-30332
Automated space vehicle control for rendezvous proximity operations	N88-30335
An automated computerized vision technique for determination of three-dimensional object geometry	N88-30348
Space truss assembly using teleoperated manipulators	N89-10087
Parallel plan execution with self-processing networks	N89-26584
On the development of a reactive sensor-based robotic system	N89-26586
BCAUS Project description and consideration of separation of data and control	N89-26591
Integration of perception and reasoning in fast neural modules	N89-26603
The procedure safety system	N90-22311
Automated procedure execution for space vehicle autonomous control	N90-22314
An architecture for object-oriented intelligent control of power systems in space	N93-25968
A linguistic geometry for space applications	N94-35050
Multiresolutional schemata for unsupervised learning of autonomous robots for 3D space operation	N94-35051
A genetic technique for planning a control sequence to navigate the state space with a quasi-minimum-cost output trajectory for a non-linear multi-dimensional system	N94-35052
Intelligent resources for satellite ground control operations	N94-35062
AUTOMATION	
Planning applications in image analysis	N94-35045
AUTONOMOUS NAVIGATION	
Robot acting on moving bodies (RAMBO): Preliminary results	N89-26597
Accurate estimation of object location in an image sequence using helicopter flight data	N94-35055
AUTONOMY	
Autonomous power expert system	N90-22306
System control of an autonomous planetary mobile spacecraft	N90-22313
Autonomous power system intelligent diagnosis and control	N91-22781
The achievement of spacecraft autonomy through the thematic application of multiple cooperating intelligent agents	N92-23364
Knowledge-based control for robot self-localization	N93-25963
An architecture for object-oriented intelligent control of power systems in space	N93-25968
An autonomous satellite architecture integrating deliberative reasoning and behavioural intelligence	N93-25970
On-board emergent scheduling of autonomous spacecraft payload operations	N94-35047
Space/ground systems as cooperating agents	N94-35072

B

BACKSCATTERING

A statistical inference approach for the retrieval of the atmospheric

ozone profile from simulated satellite measurements of solar backscattered ultraviolet radiation	N94-35060
BANDWIDTH	
Interactive archives of scientific data	N94-35044
BIOMASS	
VEG: An intelligent workbench for analysing spectral reflectance data	N94-35059
BREADBOARD MODELS	
Autonomous power system intelligent diagnosis and control	N91-22781
C	
C (PROGRAMMING LANGUAGE)	
Using C to build a satellite scheduling expert system: Examples from the Explorer platform planning system	N91-22774
C++ (PROGRAMMING LANGUAGE)	
C++ Planning and Resource Reasoning (PARR) shell	N94-35054
CABLES (ROPES)	
Cable applications in robot compliant devices	N89-10102
CAMERAS	
A connectionist model for dynamic control	N89-26604
CELESTIAL GEODESY	
Propagating orientation constraints for the Hubble Space Telescope	N94-35049
CIRCUIT DIAGRAMS	
REDEX: The ranging equipment diagnostic expert system	N89-26595
CLASSIFICATIONS	
Improved image classification with neural networks by fusing multispectral signatures with topological data	N92-23368
Combining factual and heuristic knowledge in knowledge acquisition	N92-23372
The probabilistic neural network architecture for high speed classification of remotely sensed imagery	N93-25972
Data fusion with artificial neural networks (ANN) for classification of earth surface from microwave satellite measurements	N93-25974
Classifying multispectral data by neural networks	N93-25976
Inferring heuristic classification hierarchies from natural language input	N93-25979
CLASSIFIERS	
Feature detection in satellite images using neural network technology	N92-23366
COGNITION	
A failure diagnosis and impact assessment prototype for Space Station Freedom	N91-22777
Validation and verification of expert systems	N91-22787
COLOR CODING	
A color-coded vision scheme for robotics	N91-22785
COLOR VISION	
A color-coded vision scheme for robotics	N91-22785
COLUMBUS SPACE STATION	
Candidate functions for advanced technology implementation in the Columbus mission planning environment	N88-30340
COMMAND AND CONTROL	
Automated satellite control in Ada	N88-30336
FIESTA: An operational decision aid for space network fault isolation	N89-10075
Automation of spacecraft control centers	N89-10078
An English language interface for constrained domains	N89-26581
Lessons learned in the development of the STOL intelligent tutoring system	N91-22795
COMMAND LANGUAGES	
Distributed expert systems for ground and space applications	N92-23361
COMMUNICATION NETWORKS	
A knowledge-based system with learning for computer communication network design	N90-22319
Space Communications Artificial Intelligence for Link Evaluation Terminal (SCAILET)	N91-22796
Design of neural networks for classification of remotely sensed imagery	N92-23367
Adaptive laser link reconfiguration using constraint propagation	N93-25967
Telecommunications issues of intelligent database management for group processing systems in the EOS era	N94-35067
COMPLEX SYSTEMS	
A proven knowledge-based approach to prioritizing process information	N91-22791
A linguistic geometry for space applications	N94-35050
COMPUTATIONAL GRIDS	
Artificial intelligence approach to planning the robotic assembly of large tetrahedral truss structures	N92-23357
COMPUTER AIDED DESIGN	
An autonomous satellite architecture integrating deliberative reasoning and behavioural intelligence	N93-25970
COMPUTER ASSISTED INSTRUCTION	
Lessons learned in the development of the STOL intelligent tutoring system	N91-22795
COMPUTER GRAPHICS	
Open control/display system for a telerobotics work station	N89-10089
COMPUTER NETWORKS	
A connectionist model for dynamic control	N89-26604
SURE (Science User Resource Expert): A science planning and scheduling assistant for a resource based environment	N90-22302
A knowledge-based system with learning for computer communication network design	N90-22319
Feature detection in satellite images using neural network technology	N92-23366
Group-oriented coordination models for distributed client-server computing	N94-35068
A distributed computing model for telemetry data processing	N94-35069
Engineering large-scale agent-based systems with consensus	N94-35071
COMPUTER PROGRAMMING	
Expert systems built by the Expert: An evaluation of OPS5	N89-10086
Resource Allocation Planning Helper (RALPH): Lessons learned	N90-22296
Knowledge-based and integrated monitoring and diagnosis in autonomous power systems	N90-22310
Evaluation of a proposed expert system development methodology: Two case studies	N90-22320
Techniques and implementation of the embedded rule-based expert system using Ada	N91-22788
Space Communications Artificial Intelligence for Link Evaluation Terminal (SCAILET)	N91-22796
Knowledge-based approach for generating target system specifications from a domain model	N92-23371
Model-based reasoning for system and software engineering: The Knowledge From Pictures (KFP) environment	N93-25978
An expert system prototype for aiding in the development of software functional requirements for NASA Goddard's command management system: A case study and lessons learned	N90-70240
COMPUTER PROGRAMS	
PST and PARR: Plan specification tools and a planning and resource reasoning shell for use in satellite mission planning	N89-26587
PLAN-IT-2: The next generation planning and scheduling tool	N90-22298
Space communications scheduler: A rule-based approach to adaptive deadline scheduling	N90-22300
Simulation-based reasoning about the physical propagation of fault effects	N90-22309
Automated procedure execution for space vehicle autonomous control	N90-22314
Bounded-time fault-tolerant rule-based systems	N90-22318
Optimum-AIV: A planning and scheduling system for spacecraft AIV	N91-22770
The Generic Spacecraft Analyst Assistant (GenSAA): A tool for automating spacecraft monitoring with expert systems	N91-22779

Validation and verification of expert systems	N91-22787
Detecting opportunities for parallel observations on the Hubble Space Telescope	N92-23359
Coordinating complex problem-solving among distributed intelligent agents	N92-23360
Distributed expert systems for ground and space applications	N92-23361
An application of machine learning to the organization of institutional software repositories	N93-25981
COMPUTER SYSTEMS DESIGN	
A natural language query system for Hubble Space Telescope proposal selection	N89-10083
Resource Allocation Planning Helper (RALPH): Lessons learned	N90-22296
The role of artificial intelligence techniques in scheduling systems	N90-22297
Evaluation of a proposed expert system development methodology: Two case studies	N90-22320
Engineering large-scale agent-based systems with consensus	N94-35071
COMPUTER SYSTEMS PERFORMANCE	
NASA ground terminal communication equipment automated fault isolation expert systems	N90-22305
Simulation-based reasoning about the physical propagation of fault effects	N90-22309
Evaluation of a proposed expert system development methodology: Two case studies	N90-22320
Proceedings of 1986 Conference on Artificial Intelligence Applications (NASA-TM-101873)	N90-70235
AI goes Forth	N90-70237
Multiperspective analysis and testing of expert systems	N90-70243
COMPUTER SYSTEMS PROGRAMS	
Expert systems tools for Hubble Space Telescope observation scheduling	N89-10081
Techniques and implementation of the embedded rule-based expert system using Ada	N91-22788
Multi-viewpoint clustering analysis	N93-25980
The design and application of a Transportable Inference Engine (TIE1)	N90-70241
COMPUTER TECHNIQUES	
The role of artificial intelligence techniques in scheduling systems	N90-22297
COMPUTER VISION	
An automated computerized vision technique for determination of three-dimensional object geometry	N88-30348
The 1989 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3033)	N89-26578
On the development of a reactive sensor-based robotic system	N89-26586
Robot acting on moving bodies (RAMBO): Preliminary results	N89-26597
The 1991 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3110)	N91-22769
Machine vision based teleoperation aid	N91-22784
A color-coded vision scheme for robotics	N91-22785
COMPUTERIZED SIMULATION	
The 1988 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3009)	N88-30330
Automatic mathematical modeling for real time simulation system	N88-30356
A simulation engine: Combining an expert system with a simulation language	N88-30358
Performance improvement of robots using a learning control scheme	N89-10105
The 1989 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3033)	N89-26578
Simulation-based reasoning about the physical propagation of fault effects	N90-22309
Representing functions/procedures and processes/structures for analysis of effects of failures on functions and operations	N91-22780
Expert operator's associate: A knowledge based system for spacecraft control	N91-22786
Application of artificial neural networks in hydrological modeling: A case study of runoff simulation of a Himalayan glacier basin	N94-35061
CONFERENCES	
The 1988 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3009)	N88-30330
Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics (NASA-TM-89663)	N89-10063
The 1989 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3033)	N89-26578
The 1990 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3068)	N90-22294
The 1991 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3110)	N91-22769
The 1993 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3200)	N93-25961
The 1994 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3268)	N94-35043
Proceedings of 1986 Conference on Artificial Intelligence Applications (NASA-TM-101873)	N90-70235
CONFIGURATION MANAGEMENT	
Analyzing spacecraft configurations through specialization and default reasoning	N90-22308
CONNECTORS	
A connectionist model for dynamic control	N89-26604
CONSOLES	
Metamorphoses of ONAV console operations: From prototype to real time application	N91-22794
CONSTRAINTS	
Constraint-based evaluation of sequential procedures	N90-22301
Adaptive laser link reconfiguration using constraint propagation	N93-25967
Propagating orientation constraints for the Hubble Space Telescope	N94-35049
CONTROL SYSTEMS DESIGN	
A shared-world conceptual model for integrating space station life sciences telescience operations	N88-30333
Artificial intelligence in a mission operations and satellite test environment	N88-30334
Automated space vehicle control for rendezvous proximity operations	N88-30335
Automated satellite control in Ada	N88-30336
Low level image processing techniques using the pipeline image processing engine in the flight telerobotic servicer	N88-30346
Issues, concerns, and initial implementation results for space based telerobotic control	N89-10091
A shared position/force control methodology for teleoperation	N89-10092
Multiple sensor smart robot hand with force control	N89-10093
An adaptive control scheme for a flexible manipulator	N89-10095
NASREN: Standard reference model for telerobot control	N89-10099
Computer hardware and software for robotic control	N89-10103
Teleoperated position control of a PUMA robot	N89-10104
Performance improvement of robots using a learning control scheme	N89-10105

System control of an autonomous planetary mobile spacecraft	N90-22313	Data fusion with artificial neural networks (ANN) for classification of earth surface from microwave satellite measurements
Multiresolutional schemata for unsupervised learning of autonomous robots for 3D space operation	N94-35051	
An intelligent control system for failure detection and controller reconfiguration	N94-35066	Using C to build a satellite scheduling expert system: Examples from the Explorer platform planning system
Space/ground systems as cooperating agents	N94-35072	N91-22774
CONTROL THEORY		Space Communications Artificial Intelligence for Link Evaluation Terminal (SCALET)
The 1991 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3110)	N91-22769	N91-22796
Representing functions/procedures and processes/structures for analysis of effects of failures on functions and operations	N91-22780	
CONTROLLERS		
An intelligent training system for space shuttle flight controllers	N88-30331	DATA LINKS
A university teaching simulation facility	N89-10088	Using expert systems to implement a semantic data model of a large mass storage system
COORDINATES	N91-22784	N90-22295
Machine vision based teleoperation aid	N91-22784	The application of connectionism to query planning/scheduling in intelligent user interfaces
COSMIC BACKGROUND EXPLORER SATELLITE		Using expert systems to implement a semantic data model of a large mass storage system
CLEAR: Communications Link Expert Assistance Resource	N89-10072	N90-22315
COSTS		A proven knowledge-based approach to prioritizing process information
Artificial intelligence costs, benefits, risks for selected spacecraft ground system automation scenarios	N88-30332	Design of neural networks for classification of remotely sensed imagery
CREW PROCEDURES (INFLIGHT)	N90-22309	The use of artificial intelligence techniques to improve the multiple payload integration process
Simulation-based reasoning about the physical propagation of fault effects	N90-22309	A spatial data handling system for retrieval of images by unrestricted regions of user interest
CRITERIA	N94-35048	N92-23373
A criterion autoscheduler for long range planning	N94-35048	Data exploration systems for databases
D		Logic programming and metadata specifications
DAMAGE ASSESSMENT		In search of meta-knowledge
A failure diagnosis and impact assessment prototype for Space Station Freedom	N91-22777	Automatic cataloguing and characterization of Earth science data using SE-trees
DATA ACQUISITION	N94-35048	Telecommunications issues of intelligent database management for ground processing systems in the EOS era
A criterion autoscheduler for long range planning	N94-35048	N94-35058
Mission Evaluation Room Intelligent Diagnostic and Analysis System (MIDAS)	N94-35064	
DATA BASE MANAGEMENT SYSTEMS		N94-35067
The second generation intelligent user interface for the crustal dynamics data information system	N88-30352	DATA PROCESSING
The utilization of neural nets in populating an object-oriented database	N89-26599	Spacelab data processing facility (SLDPF) Quality Assurance (QA)/Data Accounting (DA) expert systems: Transition from prototypes to operational systems
An intelligent user interface for browsing satellite data catalogs	N89-26601	N88-30353
Natural language processing and advanced information management	N89-26602	Automated cataloging and characterization of space-derived data
Telecommunications issues of intelligent database management for ground processing systems in the EOS era	N94-35067	N88-30354
DATA BASES		Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics (NASA-TM-89663)
The advice taker/inquirer: A system for high-level acquisition of expert knowledge	N88-30359	N88-10063
Data exploration systems for databases	N92-23374	An expert system approach to astronomical data analysis
Visualizing the semantic content of large text databases using text maps	N93-25982	N89-10080
The StarView intelligent query mechanism	N93-25984	Natural language processing and advanced information management
VEG: An intelligent workbench for analysing spectral reflectance data	N94-35059	N89-26602
The development of an intelligent user interface for NASA's scientific databases	N90-70236	Using C to build a satellite scheduling expert system: Examples from the Explorer platform planning system
DATA COMPRESSION	N94-35044	N91-22774
Interactive archives of scientific data	N94-35044	Improved interpretation of satellite altimeter data using genetic algorithms
An adaptive technique to maximize lossless image data compression of satellite images	N94-35057	N92-23369
DATA INTEGRATION	N89-26603	Automated analysis of complex data
Integration of perception and reasoning in fast neural modules	N89-26603	Mission Evaluation Room Intelligent Diagnostic and Analysis System (MIDAS)
		N94-35064
		ISTAR: Intelligent System for Telemetry Analysis in Real-time
		N94-35070
DATA REDUCTION		DATA STORAGE
		Autonomous image data reduction by analysis and interpretation
		N88-30347
		DATA STORAGE
		Using expert systems to implement a semantic data model of a large mass storage system
		N90-22315
		Telecommunications issues of intelligent database management for ground processing systems in the EOS era
		N94-35067
DATA STRUCTURES		Lisp Object State Saver (LOSS): A facility used to save partial

schedules of the Hubble Space Telescope	N88-30360
Knowledge structure representation and automated updates in intelligent information management systems	N90-22316
A spatial data handling system for retrieval of images by unrestricted regions of user interest	N92-23373
DATA SYSTEMS	
A spatial data handling system for retrieval of images by unrestricted regions of user interest	N92-23373
Data exploration systems for databases	N92-23374
DIOGENES: Expert system for extraction of data system requirements	N90-70245
DECISION MAKING	
The advice taker/inquirer: A system for high-level acquisition of expert knowledge	N88-30359
Verification and validation of rulebased systems for Hubble Space Telescope ground support	N88-30361
FIESTA: An operational decision aid for space network fault isolation	N89-10075
Automation of spacecraft control centers	N89-10078
A situated reasoning architecture for space-based repair and replace tasks	N89-26583
Sharing intelligence: Decision-making interactions between users and software in MAESTRO	N91-22772
A failure diagnosis and impact assessment prototype for Space Station Freedom	N91-22777
A reusable knowledge acquisition shell: KASH	N91-22789
Vista goes online: Decision-analytic systems for real-time decision-making in mission control	N94-35063
Mission Evaluation Room Intelligent Diagnostic and Analysis System (MIDAS)	N94-35064
DECISION THEORY	
Vista goes online: Decision-analytic systems for real-time decision-making in mission control	N94-35063
DEEP SPACE NETWORK	
Deep space network resource scheduling approach and application	N89-10070
DESIGN ANALYSIS	
The mission operations planning assistant	N89-10066
Robot hands and extravehicular activity	N89-10097
DETECTION	
Detection of bearing failure in mechanical devices using neural networks	N93-25965
Model-based reasoning for system and software engineering: The Knowledge From Pictures (KFP) environment	N93-25978
DIAGNOSIS	
A failure diagnosis and impact assessment prototype for Space Station Freedom	N91-22777
Autonomous power system intelligent diagnosis and control	N91-22781
A proven knowledge-based approach to prioritizing process information	N91-22791
Machine learning techniques for fault isolation and sensor placement	N93-25966
The use of multiple models in case-based diagnosis	N93-25969
Anomalous event diagnosis for environmental satellite systems	N93-25971
Qualitative model-based diagnosis using possibility theory	N94-35065
DIFFERENTIAL PULSE CODE MODULATION	
An adaptive technique to maximize lossless image data compression of satellite images	N94-35057
DISCRIMINATION	
Preliminary results of investigations into the use of artificial neural networks for discriminating gas chromatograph mass spectra of remote samples	N91-22793
DISPLAY DEVICES	
Open control/display system for a telerobotics work station	N89-10089
DISTRIBUTED PARAMETER SYSTEMS	
A hierarchical distributed control model for coordinating intelligent systems	N91-22783
DISTRIBUTED PROCESSING	
Parallel and distributed computation for fault-tolerant object recognition	N88-30350
A hierarchically distributed architecture for fault isolation expert systems on the space station	N89-10077
Advanced data management design for autonomous telerobotic systems in space using spaceborne symbolic processors	N89-10096
Coordinating complex problem-solving among distributed intelligent agents	N92-23360
Distributed expert systems for ground and space applications	N92-23361
The WorkPlace distributed processing environment	N93-25977
Group-oriented coordination models for distributed client-server computing	N94-35068
A distributed computing model for telemetry data processing	N94-35069
Engineering large-scale agent-based systems with consensus	N94-35071
DOCUMENTS	
Interactive archives of scientific data	N94-35044
DOMAINS	
Automated analysis of complex data	N94-35056
DYNAMIC CONTROL	
A connectionist model for dynamic control	N89-26604
DYNAMIC MODELS	
An adaptive control scheme for a flexible manipulator	N89-10095
E	
EARTH CRUST	
The second generation intelligent user interface for the crustal dynamics data information system	N88-30352
EARTH OBSERVATIONS (FROM SPACE)	
The utilization of neural nets in populating an object-oriented database	N89-26599
EARTH OBSERVING SYSTEM (EOS)	
SURE (Science User Resource Expert): A science planning and scheduling assistant for a resource based environment	N90-22302
In search of meta-knowledge	N93-25983
Automatic cataloguing and characterization of Earth science data using SE-trees	N94-35058
A statistical inference approach for the retrieval of the atmospheric ozone profile from simulated satellite measurements of solar backscattered ultraviolet radiation	N94-35060
Telecommunications issues of intelligent database management for ground processing systems in the EOS era	N94-35067
EARTH RADIATION BUDGET EXPERIMENT	
An expert system for scheduling requests for communications Links between TDRSS and ERBS	N89-10065
EARTH SCIENCES	
Planning applications in image analysis	N94-35045
Automatic cataloguing and characterization of Earth science data using SE-trees	N94-35058
A statistical inference approach for the retrieval of the atmospheric ozone profile from simulated satellite measurements of solar backscattered ultraviolet radiation	N94-35060
EARTH SURFACE	
Data fusion with artificial neural networks (ANN) for classification of earth surface from microwave satellite measurements	N93-25974
EDGE DETECTION	
A color-coded vision scheme for robotics	N91-22785
ELECTRICAL FAULTS	
Spacelab Life Sciences-1 electrical diagnostic expert system	N89-26593
Knowledge-based and integrated monitoring and diagnosis in autonomous power systems	N90-22310

EMBEDDED COMPUTER SYSTEMS	
Techniques and implementation of the embedded rule-based expert system using Ada	N91-22788
EMERGENCIES	
Design of an intelligent information system for in-flight emergency assistance	N91-22792
END EFFECTORS	
Multiple sensor smart robot hand with force control	N89-10093
Robot hands and extravehicular activity	N89-10097
ENGINEERS	
Mission Evaluation Room Intelligent Diagnostic and Analysis System (MIDAS)	N94-35064
ENTROPY	
An adaptive technique to maximize lossless image data compression of satellite images	N94-35057
ENVIRONMENTAL CONTROL	
Evaluating model accuracy for model-based reasoning	N92-23362
ENVIRONMENTS	
The load shedding advisor: An example of a crisis-response expert system	N89-10073
ERROR ANALYSIS	
Achieving real-time performance in FIESTA	N88-30344
An architecture for the development of real-time fault diagnosis systems using model-based reasoning	N92-23363
Intelligent fault isolation and diagnosis for communication satellite systems	N92-23365
The use of multiple models in case-based diagnosis	N93-25969
Application of artificial neural networks in hydrological modeling: A case study of runoff simulation of a Himalayan glacier basin	N94-35061
ERROR CORRECTING CODES	
Spacelab Life Sciences-1 electrical diagnostic expert system	N89-26593
Expert system development methodology and the transition from prototyping to operations: FIESTA, a case study	N89-26605
ERROR DETECTION CODES	
SHARP: A multi-mission AI system for spacecraft telemetry monitoring and diagnosis	N89-26594
Expert system development methodology and the transition from prototyping to operations: FIESTA, a case study	N89-26605
A prototype expert system in OPS5 for data error detection	N90-70242
ERRORS	
Model-based reasoning for system and software engineering: The Knowledge From Pictures (KFP) environment	N93-25978
EUROPEAN SPACE AGENCY	
Candidate functions for advanced technology implementation in the Columbus mission planning environment	N88-30340
EXPERT SYSTEMS	
The 1988 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3009)	N88-30330
An intelligent training system for space shuttle flight controllers	N88-30331
Artificial intelligence costs, benefits, risks for selected spacecraft ground system automation scenarios	N88-30332
Artificial intelligence in a mission operations and satellite test environment	N88-30334
Automated space vehicle control for rendezvous proximity operations	N88-30335
Automated satellite control in Ada	N88-30336
Knowledge based tools for Hubble Space Telescope planning and scheduling: Constraints and strategies	N88-30338
The proposal entry processor: Telescience applications for Hubble Space Telescope science operations	N88-30339
Candidate functions for advanced technology implementation in the Columbus mission planning environment	N88-30340
A rule-based systems approach to spacecraft communications configuration optimization	N88-30341
MOORE: A prototype expert system for diagnosing spacecraft problems	N88-30343
Achieving real-time performance in FIESTA	N88-30344
Mission telemetry system monitor: A real-time knowledge-based system	
Low level image processing techniques using the pipeline image processing engine in the flight telerobotic servicer	N88-30346
An interactive testbed for development of expert tools for pattern recognition	N88-30349
The second generation intelligent user interface for the crustal dynamics data information system	N88-30352
Spacelab data processing facility (SLDPF) Quality Assurance (QA)/Data Accounting (DA) expert systems: Transition from prototypes to operational systems	N88-30353
Automated cataloging and characterization of space-derived data	N88-30354
A design for a ground-based data management system	N88-30355
Automatic mathematical modeling for real time simulation system	N88-30356
A simulation engine: Combining an expert system with a simulation language	N88-30358
The advice taker/inquirer: A system for high-level acquisition of expert knowledge	N88-30359
Verification and validation of rulebased systems for Hubble Space Telescope ground support	N88-30361
The need for a comprehensive expert system development methodology	N88-30362
Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics (NASA-TM-89663)	N89-10063
Maintaining an expert system for the Hubble Space Telescope ground support	N89-10064
An expert system for scheduling requests for communications Links between TDRSS and ERBS	N89-10065
The mission operations planning assistant	N89-10066
PLAN-IT: Scheduling assistant for solar system exploration	N89-10068
An expert system that performs a satellite station keeping maneuver	N89-10069
Space station Platform Management System (PMS) replanning using resource envelopes	N89-10071
CLEAR: Communications Link Expert Assistance Resource	N89-10072
The load shedding advisor: An example of a crisis-response expert system	N89-10073
A local area computer network expert system framework	N89-10074
FIESTA: An operational decision aid for space network fault isolation	N89-10075
Expert system support for HST operations	N89-10076
A hierarchically distributed architecture for fault isolation expert systems on the space station	N89-10077
Automation of spacecraft control centers	N89-10078
Spacelab Data Processing Facility (SLDPF) quality assurance expert systems development	N89-10079
An expert system approach to astronomical data analysis	N89-10080
Expert systems tools for Hubble Space Telescope observation scheduling	N89-10081
Toward an expert project management system	N89-10082
A LISP-Ada connection	N89-10085
Expert systems built by the Expert: An evaluation of OPS5	N89-10086
Open control/display system for a telerobotics work station	N89-10089
Advanced data management design for autonomous telerobotic systems in space using spaceborne symbolic processors	N89-10096
The 1989 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3033)	N89-26578
Knowledge based and interactive control for the Superfluid Helium On-orbit Transfer Project	N89-26579
An English language interface for constrained domains	N89-26581

Ground data systems resource allocation process	N89-26582	Distributed expert systems for ground and space applications	N92-23361
An approach to knowledge engineering to support knowledge-based simulation of payload ground processing at the Kennedy Space Center	N89-26588	The achievement of spacecraft autonomy through the thematic application of multiple cooperating intelligent agents	N92-23364
BCAUS Project description and consideration of separation of data and control	N89-26591	Intelligent fault isolation and diagnosis for communication satellite systems	N92-23365
Tracking and data relay satellite fault isolation and correction using PACES: Power and attitude control expert system	N89-26592	The use of artificial intelligence techniques to improve the multiple payload integration process	N92-23370
Spacelab Life Sciences-1 electrical diagnostic expert system	N89-26593	Combining factual and heuristic knowledge in knowledge acquisition	N92-23372
REDEX: The ranging equipment diagnostic expert system	N89-26595	A spatial data handling system for retrieval of images by unrestricted regions of user interest	N92-23373
Development of an intelligent interface for adding spatial objects to a knowledge-based geographic information system	N89-26598	Using AI/expert system technology to automate planning and replanning for the HST servicing missions	N93-25962
An intelligent user interface for browsing satellite data catalogs	N89-26601	Adaptive laser link reconfiguration using constraint propagation	N93-25967
Natural language processing and advanced information management	N89-26602	The WorkPlace distributed processing environment	N93-25977
Integration of perception and reasoning in fast neural modules	N89-26603	Inferring heuristic classification hierarchies from natural language input	N93-25979
Expert system development methodology and the transition from prototyping to operations: FIESTA, a case study	N89-26605	Multi-viewpoint clustering analysis	N93-25980
Applications of fuzzy sets to rule-based expert system development	N89-26606	The 1994 Goddard Conference on Space Applications of Artificial Intelligence	
Genetic algorithms applied to the scheduling of the Hubble Space Telescope	N89-26607	(NASA-CP-3268)	N94-35043
A knowledge-based approach to improving optimization techniques in system planning	N90-22303	RAVE: Rapid Visualization Environment	N94-35046
NASA ground terminal communication equipment automated fault isolation expert systems	N90-22305	Propagating orientation constraints for the Hubble Space Telescope	N94-35049
Autonomous power expert system	N90-22306	A distributed computing model for telemetry data processing	N94-35069
Knowledge-based and integrated monitoring and diagnosis in autonomous power systems	N90-22310	ISTAR: Intelligent System for Telemetry Analysis in Real-time	N94-35070
The procedure safety system	N90-22311	Proceedings of 1986 Conference on Artificial Intelligence Applications	
Using expert systems to implement a semantic data model of a large mass storage system	N90-22315	(NASA-TM-101873)	N90-70235
Adaptive pattern recognition by mini-max neural networks as a part of an intelligent processor	N90-22317	The development of an intelligent user interface for NASA's scientific databases	N90-70236
A knowledge-based system with learning for computer communication network design	N90-22319	The Expert Project Management System (EPMS)	N90-70238
Evaluation of a proposed expert system development methodology: Two case studies	N90-22320	An expert system for ground support of the Hubble space telescope	N90-70239
Optimum-AIV: A planning and scheduling system for spacecraft AIV	N91-22770	An expert system prototype for aiding in the development of software functional requirements for NASA Goddard's command management system: A case study and lessons learned	N90-70240
TDRSS momentum unload planning	N91-22771	The design and application of a Transportable Inference Engine (TIE1)	N90-70241
Transformation reborn: A new generation expert system for planning HST operations	N91-22773	A prototype expert system in OPS5 for data error detection	N90-70242
Using C to build a satellite scheduling expert system: Examples from the Explorer platform planning system	N91-22774	Multiperspective analysis and testing of expert systems	N90-70243
The Generic Spacecraft Analyst Assistant (GenSAA): A tool for automating spacecraft monitoring with expert systems	N91-22779	Simple methods of exploiting the underlying structure of rule-based systems	N90-70244
A machine independent expert system for diagnosing environmentally induced spacecraft anomalies	N91-22782	DIOGENES: Expert system for extraction of data system requirements	N90-70245
A hierarchical distributed control model for coordinating intelligent systems	N91-22783	EXTRACTION	
Expert operator's associate: A knowledge based system for spacecraft control	N91-22786	DIOGENES: Expert system for extraction of data system requirements	N90-70245
Validation and verification of expert systems	N91-22787	EXTRAVEHICULAR ACTIVITY	
Techniques and implementation of the embedded rule-based expert system using Ada	N91-22788	Robot hands and extravehicular activity	N89-10097
A reusable knowledge acquisition shell: KASH	N91-22789		
Capturing flight system test engineering expertise: Lessons learned	N91-22790		
A proven knowledge-based approach to prioritizing process information	N91-22791		
Design of an intelligent information system for in-flight emergency assistance	N91-22792		
Metamorphoses of ONAV console operations: From prototype to real time application	N91-22794		
Lessons learned in the development of the STOL intelligent tutoring system	N91-22795		
SLS-PLAN-JT: A knowledge-based blackboard scheduling system for Spacelab life sciences missions	N92-23358		

F

F-111 AIRCRAFT	
Knowledge repositories for multiple uses	N91-22797
FAILURE	
A failure diagnosis and impact assessment prototype for Space Station Freedom	N91-22777
Representing functions/procedures and processes/structures for analysis of effects of failures on functions and operations	N91-22780
Detection of bearing failure in mechanical devices using neural networks	N93-25965
FAILURE ANALYSIS	
Expert system support for HST operations	N89-10076

The 1989 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3033)	N89-26578	On the development of a reactive sensor-based robotic system N89-26586
A method for interactive satellite failure diagnosis: Towards a connectionist solution	N89-26590	Robot acting on moving bodies (RAMBO): Preliminary results N89-26597
Tracking and data relay satellite fault isolation and correction using PACES: Power and attitude control expert system	N89-26592	Applications of fuzzy sets to rule-based expert system development N89-26606
Spacelab Life Sciences-1 electrical diagnostic expert system	N89-26593	
SHARP: A multi-mission AI system for spacecraft telemetry monitoring and diagnosis	N89-26594	
REDEX: The ranging equipment diagnostic expert system	N89-26595	
Diagnostic tolerance for missing sensor data	N89-26596	
Knowledge-based and integrated monitoring and diagnosis in autonomous power systems	N90-22310	
A failure recovery planning prototype for Space Station Freedom	N91-22778	
Qualitative model-based diagnosis using possibility theory	N94-35065	
An intelligent control system for failure detection and controller reconfiguration	N94-35066	
FAILURE MODES		
A failure recovery planning prototype for Space Station Freedom	N91-22778	
FAULT DETECTION		
An intelligent control system for failure detection and controller reconfiguration	N94-35066	
FAULT TOLERANCE		
MOORE: A prototype expert system for diagnosing spacecraft problems	N88-30343	
Achieving real-time performance in FIESTA	N88-30344	
Parallel and distributed computation for fault-tolerant object recognition	N88-30350	
Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics (NASA-TM-89663)	N89-10063	
A local area computer network expert system framework	N89-10074	
FIESTA: An operational decision aid for space network fault isolation	N89-10075	
A hierarchically distributed architecture for fault isolation expert systems on the space station	N89-10077	
Diagnostic tolerance for missing sensor data	N89-26596	
The 1990 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3068)	N90-22294	
NASA ground terminal communication equipment automated fault isolation expert systems	N90-22305	
Autonomous power expert system	N90-22306	
Adaptive pattern recognition by mini-max neural networks as a part of an intelligent processor	N90-22317	
Bounded-time fault-tolerant rule-based systems	N90-22318	
The 1992 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3141)	N92-23356	
Machine learning techniques for fault isolation and sensor placement	N93-25966	
FAULTS		
NASA ground terminal communication equipment automated fault isolation expert systems	N90-22305	
The Generic Spacecraft Analyst Assistant (GenSAA): A tool for automating spacecraft monitoring with expert systems	N91-22779	
Intelligent fault isolation and diagnosis for communication satellite systems	N92-23365	
Machine learning techniques for fault isolation and sensor placement	N93-25966	
FEEDBACK		
Consolidated fuel reprocessing program: The implications of force reflection for teleoperation in space	N89-10090	
FEEDBACK CONTROL		
Computer hardware and software for robotic control	N89-10103	
On the development of a reactive sensor-based robotic system N89-26586		
Robot acting on moving bodies (RAMBO): Preliminary results N89-26597		
Applications of fuzzy sets to rule-based expert system development N89-26606		
FLIGHT CONTROL		
An intelligent training system for space shuttle flight controllers	N88-30331	
Metamorphoses of ONAV console operations: From prototype to real time application	N91-22794	
Vista goes online: Decision-analytic systems for real-time decision-making in mission control	N94-35063	
FLIGHT OPERATIONS		
The Generic Spacecraft Analyst Assistant (GenSAA): A tool for automating spacecraft monitoring with expert systems	N91-22779	
FLIGHT TESTS		
Capturing flight system test engineering expertise: Lessons learned	N91-22790	
FORMAT		
VEG: An intelligent workbench for analysing spectral reflectance data	N94-35059	
An expert system for ground support of the Hubble space telescope	N90-70239	
FUNCTIONAL DESIGN SPECIFICATIONS		
An expert system prototype for aiding in the development of software functional requirements for NASA Goddard's command management system: A case study and lessons learned	N90-70240	
FUZZY SETS		
Applications of fuzzy sets to rule-based expert system development	N89-26606	
G		
GALILEO PROJECT		
Mission telemetry system monitor: A real-time knowledge-based system	N88-30345	
GAMMA RAY OBSERVATORY		
BCAUS Project description and consideration of separation of data and control	N89-26591	
GAS CHROMATOGRAPHY		
Preliminary results of investigations into the use of artificial neural networks for discriminating gas chromatograph mass spectra of remote samples	N91-22793	
GENETIC ALGORITHMS		
Genetic algorithms applied to the scheduling of the Hubble Space Telescope	N89-26607	
Improved interpretation of satellite altimeter data using genetic algorithms	N92-23369	
GENETICS		
Improved interpretation of satellite altimeter data using genetic algorithms	N92-23369	
GEOGRAPHIC INFORMATION SYSTEMS		
Development of an intelligent interface for adding spatial objects to a knowledge-based geographic information system	N89-26598	
GEOMETRY		
An automated computerized vision technique for determination of three-dimensional object geometry	N88-30348	
A linguistic geometry for space applications	N94-35050	
GLACIERS		
Application of artificial neural networks in hydrological modeling: A case study of runoff simulation of a Himalayan glacier basin	N94-35061	
GLOBAL POSITIONING SYSTEM		
Knowledge-based control for robot self-localization	N93-25963	
GOALS		
Representing functions/procedures and processes/structures for analysis of effects of failures on functions and operations	N91-22780	
A reusable knowledge acquisition shell: KASH	N91-22789	
GRAPHS (CHARTS)		
Model-based reasoning for system and software engineering: The Knowledge From Pictures (KFP) environment	N93-25978	

GROUND BASED CONTROL		
Intelligent resources for satellite ground control operations	N94-35062	deadline scheduling N90-22300
A distributed computing model for telemetry data processing	N94-35069	A machine independent expert system for diagnosing environmentally induced spacecraft anomalies N91-22782
Space/ground systems as cooperating agents	N94-35072	A proven knowledge-based approach to prioritizing process information N91-22791
GROUND OPERATIONAL SUPPORT SYSTEM		SLS-PLAN-IT: A knowledge-based blackboard scheduling system for Spacelab life sciences missions N92-23358
Distributed expert systems for ground and space applications	N92-23361	An architecture for the development of real-time fault diagnosis systems using model-based reasoning N92-23363
GROUND SUPPORT EQUIPMENT		Combining factual and heuristic knowledge in knowledge acquisition N92-23372
Artificial intelligence costs, benefits, risks for selected spacecraft ground system automation scenarios	N88-30332	Adaptive laser link reconfiguration using constraint propagation N93-25967
Maintaining an expert system for the Hubble Space Telescope ground support	N89-10064	Inferring heuristic classification hierarchies from natural language input N93-25979
NASA ground terminal communication equipment automated fault isolation expert systems	N90-22305	A criterion autoscheduler for long range planning N94-35048
An expert system for ground support of the Hubble space telescope	N90-70239	A linguistic geometry for space applications N94-35050
GROUND SUPPORT SYSTEMS		Automated analysis of complex data N94-35056
Expert operator's associate: A knowledge based system for spacecraft control	N91-22786	HIERARCHIES
Mission Evaluation Room Intelligent Diagnostic and Analysis System (MIDAS)	N94-35064	A hierarchically distributed architecture for fault isolation expert systems on the space station N89-10077
ISTAR: Intelligent System for Telemetry Analysis in Real-time	N94-35070	Maintaining consistency between planning hierarchies: Techniques and applications N89-10084
H		NASREN: Standard reference model for telerobot control N89-10099
HARDWARE		A hierarchical distributed control model for coordinating intelligent systems N91-22783
REDEX: The ranging equipment diagnostic expert system	N89-26595	Combining factual and heuristic knowledge in knowledge acquisition N92-23372
HELICOPTERS		Inferring heuristic classification hierarchies from natural language input N93-25979
Accurate estimation of object location in an image sequence using helicopter flight data	N94-35055	Multi-viewpoint clustering analysis N93-25980
HETEROGENEITY		A linguistic geometry for space applications N94-35050
Modeling heterogeneous processor scheduling for real time systems	N94-35053	HUBBLE SPACE TELESCOPE
HEURISTIC METHODS		Knowledge based tools for Hubble Space Telescope planning and scheduling: Constraints and strategies N88-30338
An English language interface for constrained domains	N89-26581	The proposal entry processor: Telescience applications for Hubble Space Telescope science operations N88-30339
A situated reasoning architecture for space-based repair and replace tasks	N89-26583	Loss Object State Saver (LOSS): A facility used to save partial schedules of the Hubble Space Telescope N88-30360
Parallel plan execution with self-processing networks	N89-26584	Expert system support for HST operations N89-10076
Mission scheduling	N89-26585	Expert systems tools for Hubble Space Telescope observation scheduling N89-10081
An approach to knowledge engineering to support knowledge-based simulation of payload ground processing at the Kennedy Space Center	N89-26588	Genetic algorithms applied to the scheduling of the Hubble Space Telescope N89-26607
A heuristic approach to incremental and reactive scheduling	N89-26589	Transformation reborn: A new generation expert system for planning HST operations N91-22773
A method for interactive satellite failure diagnosis: Towards a connectionist solution	N89-26590	Long range science scheduling for the Hubble Space Telescope N91-22775
Tracking and data relay satellite fault isolation and correction using PACES: Power and attitude control expert system	N89-26592	Detecting opportunities for parallel observations on the Hubble Space Telescope N92-23359
SHARP: A multi-mission AI system for spacecraft telemetry monitoring and diagnosis	N89-26594	A criterion autoscheduler for long range planning N94-35048
REDEX: The ranging equipment diagnostic expert system	N89-26595	Propagating orientation constraints for the Hubble Space Telescope N94-35049
Diagnostic tolerance for missing sensor data	N89-26596	An expert system for ground support of the Hubble space telescope N90-70239
Development of an intelligent interface for adding spatial objects to a knowledge-based geographic information system	N89-26598	HUMAN FACTORS ENGINEERING
An intelligent user interface for browsing satellite data catalogs	N89-26601	Design of an intelligent information system for in-flight emergency assistance N91-22792
Integration of perception and reasoning in fast neural modules	N89-26603	HUMAN-COMPUTER INTERFACE
Expert system development methodology and the transition from prototyping to operations: FIESTA, a case study	N89-26605	Shared resource control between human and computer N89-26580
Applications of fuzzy sets to rule-based expert system development	N89-26606	An English language interface for constrained domains N89-26581
An approach to rescheduling activities based on determination of priority and disruptivity	N90-22299	BCAUS Project description and consideration of separation of data and control N89-26591
Space communications scheduler: A rule-based approach to adaptive		Development of an intelligent interface for adding spatial objects to a knowledge-based geographic information system N89-26598
		The utilization of neural nets in populating an object-oriented database N89-26599
		A rapid prototyping/artificial intelligence approach to space station-era information management and access N89-26600

An intelligent user interface for browsing satellite data catalogs	N91-22777
The application of connectionism to query planning/scheduling in intelligent user interfaces	N89-26601
Sharing intelligence: Decision-making interactions between users and software in MAESTRO	N90-22295
Intelligent resources for satellite ground control operations	N91-22772
The development of an intelligent user interface for NASA's scientific databases	N94-35062
Simple methods of exploiting the underlying structure of rule-based systems	N90-70236
HYDRAULIC EQUIPMENT	N90-70244
Knowledge repositories for multiple uses	N91-22797
HYDROLOGY MODELS	
Application of artificial neural networks in hydrological modeling: A case study of runoff simulation of a Himalayan glacier basin	N94-35061
I	
IMAGE ANALYSIS	
Autonomous image data reduction by analysis and interpretation	N88-30347
Satellite image analysis using neural networks	N90-22321
Feature detection in satellite images using neural network technology	N92-23366
Design of neural networks for classification of remotely sensed imagery	N92-23367
Image analysis by integration of disparate information	N93-25973
Interactive archives of scientific data	N94-35044
Planning applications in image analysis	N94-35045
IMAGE CLASSIFICATION	
Improved image classification with neural networks by fusing multispectral signatures with topological data	N92-23368
IMAGE CORRELATORS	
Improved image classification with neural networks by fusing multispectral signatures with topological data	N92-23368
IMAGE PROCESSING	
The 1988 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3009)	N88-30330
Low level image processing techniques using the pipeline image processing engine in the flight telerobotic servicer	N88-30346
An interactive testbed for development of expert tools for pattern recognition	N88-30349
The 1989 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3033)	N89-26578
Satellite image analysis using neural networks	N90-22321
Feature detection in satellite images using neural network technology	N92-23366
Design of neural networks for classification of remotely sensed imagery	N92-23367
Improved image classification with neural networks by fusing multispectral signatures with topological data	N92-23368
Data fusion with artificial neural networks (ANN) for classification of earth surface from microwave satellite measurements	N93-25974
Interactive archives of scientific data	N94-35044
Planning applications in image analysis	N94-35045
Accurate estimation of object location in an image sequence using helicopter flight data	N94-35055
IMAGERY	
The probabilistic neural network architecture for high speed classification of remotely sensed imagery	N93-25972
Image analysis by integration of disparate information	N93-25973
IMAGING TECHNIQUES	
A color-coded vision scheme for robotics	N91-22785
An adaptive technique to maximize lossless image data compression of satellite images	N94-35057
IMPACT	
A failure diagnosis and impact assessment prototype for Space	
Station Freedom	N91-22777
INDEXES (DOCUMENTATION)	
A design for a ground-based data management system	N88-30355
INDUCTION	
A knowledge-based system with learning for computer communication network design	N90-22319
INFERENCE	
Analyzing spacecraft configurations through specialization and default reasoning	N90-22308
Knowledge structure representation and automated updates in intelligent information management systems	N90-22316
A machine independent expert system for diagnosing environmentally induced spacecraft anomalies	N91-22782
Logic programming and metadata specifications	N92-23375
INFORMATION MANAGEMENT	
A rapid prototyping/artificial intelligence approach to space station-era information management and access	N89-26600
Natural language processing and advanced information management	N89-26602
Using expert systems to implement a semantic data model of a large mass storage system	N90-22315
Knowledge structure representation and automated updates in intelligent information management systems	N90-22316
The 1991 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3110)	N91-22769
Automatic cataloguing and characterization of Earth science data using SE-trees	N94-35058
INFORMATION RETRIEVAL	
The utilization of neural nets in populating an object-oriented database	N89-26599
A rapid prototyping/artificial intelligence approach to space station-era information management and access	N89-26600
An intelligent user interface for browsing satellite data catalogs	N89-26601
Design of an intelligent information system for in-flight emergency assistance	N91-22792
INFORMATION SYSTEMS	
A shared-world conceptual model for integrating space station life sciences telescience operations	N88-30333
The second generation intelligent user interface for the crustal dynamics data information system	N88-30352
A LISP-Ada connection	N89-10085
Capturing flight system test engineering expertise: Lessons learned	N91-22790
Design of an intelligent information system for in-flight emergency assistance	N91-22792
A spatial data handling system for retrieval of images by unrestricted regions of user interest	N92-23373
The use of multiple models in case-based diagnosis	N93-25969
In search of meta-knowledge	N93-25983
INFORMATION THEORY	
Qualitative model-based diagnosis using possibility theory	N94-35065
INFORMATION TRANSFER	
Coordinating complex problem-solving among distributed intelligent agents	N92-23360
INSPECTION	
The JPL/KSC telerobotic inspection demonstration	N90-22312
INSTRUMENTS	
The mission operations planning assistant	N89-10066
INTEGRATED MISSION CONTROL CENTER	
A distributed computing model for telemetry data processing	N94-35069
INTERACTIVE CONTROL	
Knowledge based and interactive control for the Superfluid Helium On-orbit Transfer Project	N89-26579
INTERPLANETARY SPACECRAFT	
System control of an autonomous planetary mobile spacecraft	N90-22313
INTERPROCESSOR COMMUNICATION	
Teleoperated position control of a PUMA robot	N89-10104

Anomalous event diagnosis for environmental satellite systems	N93-25971	Combining factual and heuristic knowledge in knowledge acquisition	N92-23372
ITERATION		The 1994 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3268)	N94-35043
Improved interpretation of satellite altimeter data using genetic algorithms	N92-23369		
RAVE: Rapid Visualization Environment	N94-35046		
K			
KERNEL FUNCTIONS			
Optimum-AIV: A planning and scheduling system for spacecraft AIV	N91-22770		
KINEMATICS			
An optimal resolved rate law for kinematically redundant manipulators	N89-10094		
Kinematic study of flight telerobotic servicer configuration issues	N89-10100		
KNOWLEDGE BASED SYSTEMS			
Multi-viewpoint clustering analysis	N93-25980		
Engineering large-scale agent-based systems with consensus	N94-35071		
KNOWLEDGE BASES (ARTIFICIAL INTELLIGENCE)			
Knowledge based and interactive control for the Superfluid Helium On-orbit Transfer Project	N89-26579		
PST and PARR: Plan specification tools and a planning and resource reasoning shell for use in satellite mission planning	N89-26587		
An approach to knowledge engineering to support knowledge-based simulation of payload ground processing at the Kennedy Space Center	N89-26588		
BCAUS Project description and consideration of separation of data and control	N89-26591		
Development of an intelligent interface for adding spatial objects to a knowledge-based geographic information system	N89-26598		
An intelligent user interface for browsing satellite data catalogs	N89-26601		
Integration of perception and reasoning in fast neural modules	N89-26603		
Genetic algorithms applied to the scheduling of the Hubble Space Telescope	N89-26607		
PLAN-IT-2: The next generation planning and scheduling tool	N90-22298		
SURE (Science User Resource Expert): A science planning and scheduling assistant for a resource based environment	N90-22302		
A knowledge-based approach to improving optimization techniques in system planning	N90-22303		
Knowledge-based and integrated monitoring and diagnosis in autonomous power systems	N90-22310		
Capturing flight system test engineering expertise: Lessons learned	N91-22790		
Knowledge repositories for multiple uses	N91-22797		
Detecting opportunities for parallel observations on the Hubble Space Telescope	N92-23359		
RAVE: Rapid Visualization Environment	N94-35046		
KNOWLEDGE REPRESENTATION			
An English language interface for constrained domains	N89-26581		
An approach to knowledge engineering to support knowledge-based simulation of payload ground processing at the Kennedy Space Center	N89-26588		
BCAUS Project description and consideration of separation of data and control	N89-26591		
Spacelab Life Sciences-1 electrical diagnostic expert system	N89-26593		
REDEX: The ranging equipment diagnostic expert system	N89-26595		
Knowledge structure representation and automated updates in intelligent information management systems	N90-22316		
The 1991 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3110)	N91-22769		
L			
LANDSAT 4			
Classifying multispectral data by neural networks	N93-25976		
LANGUAGES			
A linguistic geometry for space applications	N94-35050		
LARGE SPACE STRUCTURES			
Artificial intelligence approach to planning the robotic assembly of large tetrahedral truss structures	N92-23357		
LASERS			
Adaptive laser link reconfiguration using constraint propagation	N93-25967		
LEAF AREA INDEX			
VEG: An intelligent workbench for analysing spectral reflectance data	N94-35059		
LEARNING			
Performance improvement of robots using a learning control scheme	N89-10105		
LEARNING CURVES			
Space Communications Artificial Intelligence for Link Evaluation Terminal (SCAILET)	N91-22796		
LIFE SCIENCES			
A shared-world conceptual model for integrating space station life sciences telescience operations	N88-30333		
LIFE SUPPORT SYSTEMS			
Evaluating model accuracy for model-based reasoning	N92-23362		
LINGUISTICS			
A linguistic geometry for space applications	N94-35050		
LIQUID HELIUM			
Knowledge based and interactive control for the Superfluid Helium On-orbit Transfer Project	N89-26579		
LISP (PROGRAMMING LANGUAGE)			
Lisp Object State Saver (LOSS): A facility used to save partial schedules of the Hubble Space Telescope	N88-30360		
A LISP-Ada connection	N89-10085		
Using C to build a satellite scheduling expert system: Examples from the Explorer platform planning system	N91-22774		
LOADS (FORCES)			
Consolidated fuel reprocessing program: The implications of force reflection for teleoperation in space	N89-10090		
A shared position/force control methodology for teleoperation	N89-10092		
Multiple sensor smart robot hand with force control	N89-10093		
LOCAL AREA NETWORKS			
A local area computer network expert system framework	N89-10074		
LOGIC PROGRAMMING			
Logic programming and metadata specifications	N92-23375		
LOW ALTITUDE			
Accurate estimation of object location in an image sequence using helicopter flight data	N94-35055		
M			
MACHINE LEARNING			
Performance improvement of robots using a learning control scheme	N89-10105		
Data exploration systems for databases	N92-23374		
Machine learning techniques for fault isolation and sensor placement	N93-25966		
An application of machine learning to the organization of institutional software repositories	N93-25981		
Multiresolutional schemata for unsupervised learning of autonomous			

robots for 3D space operation	N94-35051
Automatic cataloguing and characterization of Earth science data using SE-trees	N94-35058
MAINTENANCE	
Maintaining an expert system for the Hubble Space Telescope ground support	N89-10064
REDEX: The ranging equipment diagnostic expert system	N89-26595
Anomalous event diagnosis for environmental satellite systems	N93-25971
MALFUNCTIONS	
Design of an intelligent information system for in-flight emergency assistance	N91-22792
MAN MACHINE SYSTEMS	
Artificial intelligence in a mission operations and satellite test environment	N88-30334
Integrated resource scheduling in a distributed scheduling environment	N88-30342
Dynamic task allocation for a man-machine symbiotic system	N89-10098
MANAGEMENT METHODS	
A failure recovery planning prototype for Space Station Freedom	N91-22778
MANAGEMENT SYSTEMS	
Design of neural networks for classification of remotely sensed imagery	N92-23367
The use of artificial intelligence techniques to improve the multiple payload integration process	N92-23370
The Expert Project Management System (EPMS)	N90-70238
An expert system prototype for aiding in the development of software functional requirements for NASA Goddard's command management system: A case study and lessons learned	N90-70240
MANIPULATORS	
Consolidated fuel reprocessing program: The implications of force reflection for teleoperation in space	N89-10090
Multiple sensor smart robot hand with force control	N89-10093
An optimal resolved rate law for kinematically redundant manipulators	N89-10094
An adaptive control scheme for a flexible manipulator	N89-10095
Robot hands and extravehicular activity	N89-10097
Actuators for a space manipulator	N89-10101
The JPL/KSC telerobotic inspection demonstration	N90-22312
Parametric motion control of robotic arms: A biologically based approach using neural networks	N93-25964
MARKOV PROCESSES	
Range data description based on multiple characteristics	N88-30351
MARS SAMPLE RETURN MISSIONS	
Knowledge-based control for robot self-localization	N93-25963
MASS SPECTRA	
Preliminary results of investigations into the use of artificial neural networks for discriminating gas chromatograph mass spectra of remote samples	N91-22793
MASSIVELY PARALLEL PROCESSORS	
Image analysis by integration of disparate information	N93-25973
MATERIALS HANDLING	
Knowledge based and interactive control for the Superfluid Helium On-orbit Transfer Project	N89-26579
MATHEMATICAL MODELS	
Range data description based on multiple characteristics	N88-30351
Automatic mathematical modeling for real time simulation system	N88-30356
A method for interactive satellite failure diagnosis: Towards a connectionist solution	N89-26590
A connectionist model for dynamic control	N89-26604
Representing functions/procedures and processes/structures for analysis of effects of failures on functions and operations	N91-22780
Evaluating model accuracy for model-based reasoning	N92-23362
MATHEMATICAL PROGRAMMING	
A knowledge-based approach to improving optimization techniques in system planning	N90-22303
MAXIMUM LIKELIHOOD ESTIMATES	
Automatic cataloguing and characterization of Earth science data using SE-trees	N94-35058
MECHANICAL DEVICES	
Detection of bearing failure in mechanical devices using neural networks	N93-25965
MESSAGE PROCESSING	
Integrated resource scheduling in a distributed scheduling environment	N88-30342
A knowledge-based system with learning for computer communication network design	N90-22319
MICROGRAVITY	
Knowledge based and interactive control for the Superfluid Helium On-orbit Transfer Project	N89-26579
MISSION PLANNING	
The 1988 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3009)	N88-30330
Contingency rescheduling of spacecraft operations	N88-30337
Knowledge based tools for Hubble Space Telescope planning and scheduling: Constraints and strategies	N88-30338
Candidate functions for advanced technology implementation in the Columbus mission planning environment	N88-30340
A rule-based systems approach to spacecraft communications configuration optimization	N88-30341
Integrated resource scheduling in a distributed scheduling environment	N88-30342
The mission operations planning assistant	N89-10066
AUTOPLAN: A PC-based automated mission planning tool	N89-10067
Expert systems tools for Hubble Space Telescope observation scheduling	N89-10081
Maintaining consistency between planning hierarchies: Techniques and applications	N89-10084
The 1989 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3033)	N89-26578
Ground data systems resource allocation process	N89-26582
Mission scheduling	N89-26585
PST and PARR: Plan specification tools and a planning and resource reasoning shell for use in satellite mission planning	N89-26587
A heuristic approach to incremental and reactive scheduling	N89-26589
AI techniques for a space application scheduling problem	N91-22776
Using AI/expert system technology to automate planning and replanning for the HST servicing missions	N93-25962
MODELS	
A shared-world conceptual model for integrating space station life sciences telescience operations	N88-30333
MOMENTUM	
TDRSS momentum unload planning	N91-22771
MONITORS	
The load shedding advisor: An example of a crisis-response expert system	N89-10073
MOVING TARGET INDICATORS	
Robot acting on moving bodies (RAMBO): Preliminary results	N89-26597
MULTIPROCESSING (COMPUTERS)	
Modeling heterogeneous processor scheduling for real time systems	N94-35053

N

NAP-OF-THE-EARTH NAVIGATION

Accurate estimation of object location in an image sequence using helicopter flight data
N94-35055

NASA PROGRAMS

The Generic Spacecraft Analyst Assistant (GenSAA): A tool for automating spacecraft monitoring with expert systems
N91-22779

NATURAL LANGUAGE (COMPUTERS)

A natural language query system for Hubble Space Telescope proposal selection	N89-10083
An English language interface for constrained domains	N89-26581
An intelligent user interface for browsing satellite data catalogs	N89-26601
Natural language processing and advanced information management	N89-26602
Logic programming and metadata specifications	N92-23375
Inferring heuristic classification hierarchies from natural language input	N93-25979
The development of an intelligent user interface for NASA's scientific databases	N90-70236
NAVIGATION AIDS	
Metamorphoses of ONAV console operations: From prototype to real time application	N91-22794
NETWORK CONTROL	
Group-oriented coordination models for distributed client-server computing	N94-35068
NEURAL NETS	
Autonomous image data reduction by analysis and interpretation	N88-30347
Parallel and distributed computation for fault-tolerant object recognition	N88-30350
Mission scheduling	N89-26585
The utilization of neural nets in populating an object-oriented database	N89-26599
Integration of perception and reasoning in fast neural modules	N89-26603
Genetic algorithms applied to the scheduling of the Hubble Space Telescope	N89-26607
An approach to rescheduling activities based on determination of priority and disruptivity	N90-22299
Adaptive pattern recognition by mini-max neural networks as a part of an intelligent processor	N90-22317
Satellite image analysis using neural networks	N90-22321
The 1991 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3110)	N91-22769
Long range science scheduling for the Hubble Space Telescope	N91-22775
Preliminary results of investigations into the use of artificial neural networks for discriminating gas chromatograph mass spectra of remote samples	N91-22793
The 1992 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3141)	N92-23356
Feature detection in satellite images using neural network technology	N92-23366
Design of neural networks for classification of remotely sensed imagery	N92-23367
Improved image classification with neural networks by fusing multispectral signatures with topological data	N92-23368
Parametric motion control of robotic arms: A biologically based approach using neural networks	N93-25964
Detection of bearing failure in mechanical devices using neural networks	N93-25965
The probabilistic neural network architecture for high speed classification of remotely sensed imagery	N93-25972
Data fusion with artificial neural networks (ANN) for classification of earth surface from microwave satellite measurements	N93-25974
Neural networks for atmospheric retrievals	N93-25975
Classifying multispectral data by neural networks	N93-25976
The 1994 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3268)	N94-35043
Application of artificial neural networks in hydrological modeling: A case study of runoff simulation of a Himalayan glacier basin	N94-35061
An intelligent control system for failure detection and controller reconfiguration	N94-35066

NOISE TOLERANCE

Machine learning techniques for fault isolation and sensor placement	N93-25966
--	-----------

NONLINEAR SYSTEMS

A genetic technique for planning a control sequence to navigate the state space with a quasi-minimum-cost output trajectory for a nonlinear multi-dimensional system	N94-35052
--	-----------

NUMERICAL CONTROL

Issues, concerns, and initial implementation results for space based telerobotic control	N89-10091
A shared position/force control methodology for teleoperation	N89-10092

O**OBJECT PROGRAMS**

Robot acting on moving bodies (RAMBO): Preliminary results	N89-26597
--	-----------

The utilization of neural nets in populating an object-oriented database	N89-26599
--	-----------

A hierarchical distributed control model for coordinating intelligent systems	N91-22783
---	-----------

OBJECT-ORIENTED PROGRAMMING

Coordinating complex problem-solving among distributed intelligent agents	N92-23360
---	-----------

C++ Planning and Resource Reasoning (PARR) shell	N94-35054
--	-----------

OBSERVATION

Transformation reborn: A new generation expert system for planning HST operations	N91-22773
---	-----------

OBSTACLE AVOIDANCE

A genetic technique for planning a control sequence to navigate the state space with a quasi-minimum-cost output trajectory for a nonlinear multi-dimensional system	N94-35052
--	-----------

Accurate estimation of object location in an image sequence using helicopter flight data	N94-35055
--	-----------

OCEAN MODELS

Improved interpretation of satellite altimeter data using genetic algorithms	N92-23369
--	-----------

OCEAN SURFACE

Improved interpretation of satellite altimeter data using genetic algorithms	N92-23369
--	-----------

ON-LINE SYSTEMS

Expert operator's associate: A knowledge based system for spacecraft control	N91-22786
--	-----------

Vista goes online: Decision-analytic systems for real-time decision-making in mission control	N94-35063
---	-----------

ONBOARD DATA PROCESSING

Spacelab Data Processing Facility (SLDPF) quality assurance expert systems development	N89-10079
--	-----------

OPERATING SYSTEMS (COMPUTERS)

A rule-based systems approach to spacecraft communications configuration optimization	N88-30341
---	-----------

Open control/display system for a telerobotics work station	N89-10089
---	-----------

Proceedings of 1986 Conference on Artificial Intelligence Applications (NASA-TM-101873)	N90-70235
---	-----------

Simple methods of exploiting the underlying structure of rule-based systems	N90-70244
---	-----------

OPERATIONAL PROBLEMS

BCAUS Project description and consideration of separation of data and control	N89-26591
---	-----------

OPERATIONS RESEARCH

Ground data systems resource allocation process	N89-26582
---	-----------

Mission scheduling	N89-26585
--------------------	-----------

SLS-PLAN-IT: A knowledge-based blackboard scheduling system for Spacelab life sciences missions	N92-23358
---	-----------

OPERATOR PERFORMANCE

A university teaching simulation facility	N89-10088
---	-----------

Lessons learned in the development of the STOL intelligent tutoring system	N91-22795
--	-----------

OPTICAL COMMUNICATION		
Adaptive laser link reconfiguration using constraint propagation		
N93-25967		
OPTIMAL CONTROL		
A genetic technique for planning a control sequence to navigate the state space with quasi-minimum-cost output trajectory for a non-linear multi-dimensional system		
N94-35052		
OPTIMIZATION		
A rule-based systems approach to spacecraft communications configuration optimization	N88-30341	
A heuristic approach to incremental and reactive scheduling	N89-26589	
A knowledge-based approach to improving optimization techniques in system planning	N90-22303	
AI techniques for a space application scheduling problem	N91-22776	
ORBITAL ASSEMBLY		
Artificial intelligence approach to planning the robotic assembly of large tetrahedral truss structures	N92-23357	
ORBITAL SERVICING		
A situated reasoning architecture for space-based repair and replace tasks	N89-26583	
OZONE		
A statistical inference approach for the retrieval of the atmospheric ozone profile from simulated satellite measurements of solar backscattered ultraviolet radiation	N94-35060	
P		
PARALLEL PROCESSING (COMPUTERS)		
Parallel and distributed computation for fault-tolerant object recognition	N88-30350	
Parallel plan execution with self-processing networks	N89-26584	
A connectionist model for dynamic control	N89-26604	
Interactive archives of scientific data	N94-35044	
PARAMETER IDENTIFICATION		
Autonomous power expert system	N90-22306	
PATTERN RECOGNITION		
An interactive testbed for development of expert tools for pattern recognition	N88-30349	
Parallel and distributed computation for fault-tolerant object recognition	N88-30350	
Adaptive pattern recognition by mini-max neural networks as a part of an intelligent processor	N90-22317	
Satellite image analysis using neural networks	N90-22321	
A color-coded vision scheme for robotics	N91-22785	
Preliminary results of investigations into the use of artificial neural networks for discriminating gas chromatograph mass spectra of remote samples	N91-22793	
Feature detection in satellite images using neural network technology	N92-23366	
PAYOUT INTEGRATION		
The use of artificial intelligence techniques to improve the multiple payload integration process	N92-23370	
PAYOUT INTEGRATION PLAN		
An approach to knowledge engineering to support knowledge-based simulation of payload ground processing at the Kennedy Space Center	N89-26588	
PAYLOADS		
AUTOPLAN: A PC-based automated mission planning tool	N89-10067	
The use of artificial intelligence techniques to improve the multiple payload integration process	N92-23370	
On-board emergent scheduling of autonomous spacecraft payload operations	N94-35047	
PERSONAL COMPUTERS		
AUTOPLAN: A PC-based automated mission planning tool	N89-10067	
PILOT PERFORMANCE		
Applications of fuzzy sets to rule-based expert system development	N89-26606	
PIXELS		
Automatic cataloguing and characterization of Earth science data using SE-trees	N94-35058	
PLANNING		
Lisp Object State Saver (LOSS): A facility used to save partial schedules of the Hubble Space Telescope	N88-30360	
Shared resource control between human and computer	N89-26580	
The 1990 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3068)	N90-22294	
The application of connectionism to query planning/scheduling in intelligent user interfaces	N90-22295	
SURE (Science User Resource Expert): A science planning and scheduling assistant for a resource based environment	N90-22302	
A knowledge-based approach to improving optimization techniques in system planning	N90-22303	
Optimum-AIV: A planning and scheduling system for spacecraft AIV	N91-22770	
TDRSS momentum unload planning	N91-22771	
Transformation reborn: A new generation expert system for planning HST operations	N91-22773	
A failure recovery planning prototype for Space Station Freedom	N91-22778	
Planning applications in image analysis	N94-35045	
PREDICTION ANALYSIS TECHNIQUES		
PST and PARR: Plan specification tools and a planning and resource reasoning shell for use in satellite mission planning	N89-26587	
Evaluating model accuracy for model-based reasoning	N92-23362	
PROBLEM SOLVING		
MOORE: A prototype expert system for diagnosing spacecraft problems	N88-30343	
The load shedding advisor: An example of a crisis-response expert system	N89-10073	
A method for interactive satellite failure diagnosis: Towards a connectionist solution	N89-26590	
Space Communications Artificial Intelligence for Link Evaluation Terminal (SCAILET)	N91-22796	
Knowledge repositories for multiple uses	N91-22797	
Combining factual and heuristic knowledge in knowledge acquisition	N92-23372	
Intelligent resources for satellite ground control operations	N94-35062	
PROGRAM VERIFICATION (COMPUTERS)		
Verification and validation of rulebased systems for Hubble Space Telescope ground support	N88-30361	
Spacecraft command verification: The AI solution	N90-22307	
Optimum-AIV: A planning and scheduling system for spacecraft AIV	N91-22770	
Validation and verification of expert systems	N91-22787	
Multiperspective analysis and testing of expert systems	N90-70243	
PROGRAMMING LANGUAGES		
AI goes Forth	N90-70237	
A prototype expert system in OPS5 for data error detection	N90-70242	
PROJECT MANAGEMENT		
Deep space network resource scheduling approach and application	N89-10070	
Toward an expert project management system	N89-10082	
The Generic Spacecraft Analyst Assistant (GenSAA): A tool for automating spacecraft monitoring with expert systems	N91-22779	
The Expert Project Management System (EPMS)	N90-70238	
PROJECT PLANNING		
Parallel plan execution with self-processing networks	N89-26584	
PROLOG (PROGRAMMING LANGUAGE)		
The load shedding advisor: An example of a crisis-response expert system	N89-10073	
PROPELLANT TRANSFER		
Knowledge based and interactive control for the Superfluid Helium On-orbit Transfer Project	N89-26579	

PROTOCOL (COMPUTERS)

- A local area computer network expert system framework N89-10074
 Group-oriented coordination models for distributed client-server computing N94-35068
 A distributed computing model for telemetry data processing N94-35069

PROTOTYPES

- Contingency rescheduling of spacecraft operations N88-30337
 Achieving real-time performance in FIESTA N88-30344
 An expert system approach to astronomical data analysis N89-10080
 Toward an expert project management system N89-10082
 Genetic algorithms applied to the scheduling of the Hubble Space Telescope N89-26607
 A failure recovery planning prototype for Space Station Freedom N91-22778
 Expert operator's associate: A knowledge based system for spacecraft control N91-22786
 Validation and verification of expert systems N91-22787
 Metamorphoses of ONAV console operations: From prototype to real time application N91-22794
 Knowledge-based approach for generating target system specifications from a domain model N92-23371
 An expert system prototype for aiding in the development of software functional requirements for NASA Goddard's command management system: A case study and lessons learned N90-70240

Q**QUALITY CONTROL**

- Spacelab Data Processing Facility (SLDPF) quality assurance expert systems development N89-10079

QUERY LANGUAGES

- A natural language query system for Hubble Space Telescope proposal selection N89-10083
 The StarView intelligent query mechanism N93-25984

R**RADAR IMAGERY**

- Range data description based on multiple characteristics N88-30351

RADIANCE

- Neural networks for atmospheric retrievals N93-25975

RANGE FINDERS

- REDEX: The ranging equipment diagnostic expert system N89-26595

RANGEFINDING

- Range data description based on multiple characteristics N88-30351

REAL TIME OPERATION

- Achieving real-time performance in FIESTA N88-30344
 Mission telemetry system monitor: A real-time knowledge-based system N88-30345
 Autonomous image data reduction by analysis and interpretation N88-30347
 Automatic mathematical modeling for real time simulation system N88-30356
 Spacelab Life Sciences-1 electrical diagnostic expert system N89-26593
 Expert system development methodology and the transition from prototyping to operations: FIESTA, a case study N89-26605
 The application of connectionism to query planning/scheduling in intelligent user interfaces N90-22295
 The Generic Spacecraft Analyst Assistant (GenSAA): A tool for automating spacecraft monitoring with expert systems N91-22779
 Metamorphoses of ONAV console operations: From prototype to real time application N91-22794
 An architecture for the development of real-time fault diagnosis

systems using model-based reasoning

N92-23363

The WorkPlace distributed processing environment N93-25977

Modeling heterogeneous processor scheduling for real time systems N94-35053

- Vista goes online: Decision-analytic systems for real-time decision-making in mission control N94-35063
 ISTAR: Intelligent System for Telemetry Analysis in Real-time N94-35070

RECOVERY

- A failure recovery planning prototype for Space Station Freedom N91-22778

REFLECTION

- Consolidated fuel reprocessing program: The implications of force reflection for teleoperation in space N89-10090

REMOTE CONTROL

- Space truss assembly using teleoperated manipulators N89-10087

REMOTE SENSING

- Diagnostic tolerance for missing sensor data N89-26596

- Design of neural networks for classification of remotely sensed imagery N92-23367

- Improved image classification with neural networks by fusing multispectral signatures with topological data N92-23368

- The probabilistic neural network architecture for high speed classification of remotely sensed imagery N93-25972

- Image analysis by integration of disparate information N93-25973

- Automatic cataloguing and characterization of Earth science data using SE-trees N94-35058

- VEG: An intelligent workbench for analysing spectral reflectance data N94-35059

REMOTE SENSORS

- A spatial data handling system for retrieval of images by unrestricted regions of user interest N92-23373

REQUIREMENTS

- Achieving real-time performance in FIESTA N88-30344

- Actuators for a space manipulator N89-10101

RESOURCE ALLOCATION

- Ground data systems resource allocation process N89-26582

- PST and PARR: Plan specification tools and a planning and resource reasoning shell for use in satellite mission planning N89-26587

- Resource Allocation Planning Helper (RALPH): Lessons learned N90-22296

RESOURCES MANAGEMENT

- Space station Platform Management System (PMS) replanning using resource envelopes N89-10071

- Automation of spacecraft control centers N89-10078

- Shared resource control between human and computer N89-26580

RESPONSE TIME (COMPUTERS)

- Bounded-time fault-tolerant rule-based systems N90-22318

REUSE

- Using C to build a satellite scheduling expert system: Examples from the Explorer platform planning system N91-22774

- A reusable knowledge acquisition shell: KASH N91-22789

RISK

- Validation and verification of expert systems N91-22787

ROBOT ARMS

- The JPL/KSC telerobotic inspection demonstration N90-22312

- Parametric motion control of robotic arms: A biologically based approach using neural networks N93-25964

ROBOT CONTROL

- Open control/display system for a telerobotics work station N89-10089

- Issues, concerns, and initial implementation results for space based telerobotic control N89-10091

- A shared position/force control methodology for teleoperation N89-10092

- Multiple sensor smart robot hand with force control N89-10093

- NASREN: Standard reference model for telerobot control N89-10099

- Computer hardware and software for robotic control N89-10103

- Teleoperated position control of a PUMA robot N89-10104

- Performance improvement of robots using a learning control scheme N89-10105

S

Parametric motion control of robotic arms: A biologically based approach using neural networks	N93-25964
A linguistic geometry for space applications	N94-35050
Multiresolutional schemata for unsupervised learning of autonomous robots for 3D space operation	N94-35051
A genetic technique for planning a control sequence to navigate the state space with a quasi-minimum-cost output trajectory for a non-linear multi-dimensional system	N94-35052
ROBOT SENSORS	
The procedure safety system	N90-22311
ROBOTICS	
Low level image processing techniques using the pipeline image processing engine in the flight telerobotic servicer	N88-30346
An automated computerized vision technique for determination of three-dimensional object geometry	N88-30348
Proceedings of 1987 Goddard Conference on Space Applications of Artificial Intelligence (AI) and Robotics (NASA-TM-89663)	N89-10063
Space truss assembly using teleoperated manipulators	N89-10087
A university teaching simulation facility	N89-10088
Open control/display system for a telerobotics work station	N89-10089
Issues, concerns, and initial implementation results for space based telerobotic control	N89-10091
A shared position/force control methodology for teleoperation	N89-10092
Multiple sensor smart robot hand with force control	N89-10093
Advanced data management design for autonomous telerobotic systems in space using spaceborne symbolic processors	N89-10096
NASREN: Standard reference model for telerobot control	N89-10099
Kinematic study of flight telerobotic servicer configuration issues	N89-10100
Actuators for a space manipulator	N89-10101
Cable applications in robot compliant devices	N89-10102
Computer hardware and software for robotic control	N89-10103
A situated reasoning architecture for space-based repair and replace tasks	N89-26583
On the development of a reactive sensor-based robotic system	N89-26586
Robot acting on moving bodies (RAMBO): Preliminary results	N89-26597
The procedure safety system	N90-22311
The JPL/KSC telerobotic inspection demonstration	N90-22312
The 1991 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3110)	N91-22769
Artificial intelligence approach to planning the robotic assembly of large tetrahedral truss structures	N92-23357
Knowledge-based control for robot self-localization	N93-25963
An autonomous satellite architecture integrating deliberative reasoning and behavioural intelligence	N93-25970
A linguistic geometry for space applications	N94-35050
ROBOTS	
An optimal resolved rate law for kinematically redundant manipulators	N89-10094
Teleoperated position control of a PUMA robot	N89-10104
Performance improvement of robots using a learning control scheme	N89-10105
Machine vision based teleoperation aid	N91-22784
A color-coded vision scheme for robotics	N91-22785
Knowledge-based control for robot self-localization	N93-25963
Parametric motion control of robotic arms: A biologically based approach using neural networks	N93-25964
Multiresolutional schemata for unsupervised learning of autonomous robots for 3D space operation	N94-35051
ROBUSTNESS (MATHEMATICS)	
The WorkPlace distributed processing environment	N93-25977
ROLLER BEARINGS	
Detection of bearing failure in mechanical devices using neural networks	N93-25965
SAFETY	
The procedure safety system	N90-22311
SATELLITE ALTIMETRY	
Improved interpretation of satellite altimeter data using genetic algorithms	N92-23369
SATELLITE COMMUNICATION	
An expert system for scheduling requests for communications Links between TDRSS and ERBS	N89-10065
Space Communications Artificial Intelligence for Link Evaluation Terminal (SCAILET)	N91-22796
Design of neural networks for classification of remotely sensed imagery	N92-23367
Anomalous event diagnosis for environmental satellite systems	N93-25971
An adaptive technique to maximize lossless image data compression of satellite images	N94-35057
SATELLITE CONTROL	
An expert system that performs a satellite station keeping maneuver	N89-10069
SATELLITE IMAGERY	
Satellite image analysis using neural networks	N90-22321
Feature detection in satellite images using neural network technology	N92-23366
Design of neural networks for classification of remotely sensed imagery	N92-23367
Improved image classification with neural networks by fusing multispectral signatures with topological data	N92-23368
An adaptive technique to maximize lossless image data compression of satellite images	N94-35057
Automatic cataloguing and characterization of Earth science data using SE-trees	N94-35058
A statistical inference approach for the retrieval of the atmospheric ozone profile from simulated satellite measurements of solar backscattered ultraviolet radiation	N94-35060
The development of an intelligent user interface for NASA's scientific databases	N90-70236
SATELLITE OBSERVATION	
Detecting opportunities for parallel observations on the Hubble Space Telescope	N92-23359
Data fusion with artificial neural networks (ANN) for classification of earth surface from microwave satellite measurements	N93-25974
A statistical inference approach for the retrieval of the atmospheric ozone profile from simulated satellite measurements of solar backscattered ultraviolet radiation	N94-35060
SATELLITE ORBITS	
An expert system that performs a satellite station keeping maneuver	N89-10069
SATELLITE ORIENTATION	
Propagating orientation constraints for the Hubble Space Telescope	N94-35049
SATELLITE TRACKING	
TDRSS momentum unload planning	N91-22771
SCHEDULING	
Contingency rescheduling of spacecraft operations	N88-30337
Knowledge based tools for Hubble Space Telescope planning and scheduling: Constraints and strategies	N88-30338
The proposal entry processor: Telescience applications for Hubble Space Telescope science operations	N88-30339
Candidate functions for advanced technology implementation in the Columbus mission planning environment	N88-30340
Integrated resource scheduling in a distributed scheduling environment	N88-30342
Lisp Object State Saver (LOSS): A facility used to save partial schedules of the Hubble Space Telescope	N88-30360
An expert system for scheduling requests for communications Links between TDRSS and ERBS	N89-10065
PLAN-IT: Scheduling assistant for solar system exploration	N89-10068
Deep space network resource scheduling approach and application	N89-10070

Space station Platform Management System (PMS) replanning using resource envelopes	N89-10071	SENSITIVITY	Neural networks for atmospheric retrievals	N93-25975
Expert systems tools for Hubble Space Telescope observation scheduling	N89-10081	SEQUENCING	Constraint-based evaluation of sequential procedures	N90-22301
Maintaining consistency between planning hierarchies: Techniques and applications	N89-10084		Spacecraft command verification: The AI solution	N90-22307
Ground data systems resource allocation process	N89-26582		Coordinating complex problem-solving among distributed intelligent agents	N92-23360
Mission scheduling	N89-26585	SEQUENTIAL CONTROL	Dynamic task allocation for a man-machine symbiotic system	N89-10098
PST and PARR: Plan specification tools and a planning and resource reasoning shell for use in satellite mission planning	N89-26587		A genetic technique for planning a control sequence to navigate the state space with a quasi-minimum-cost output trajectory for a non-linear multi-dimensional system	N94-35052
An approach to knowledge engineering to support knowledge-based simulation of payload ground processing at the Kennedy Space Center	N89-26588	SERIES EXPANSION	Adaptive pattern recognition by mini-max neural networks as a part of an intelligent processor	N90-22317
A heuristic approach to incremental and reactive scheduling	N89-26589		SIGNAL PROCESSING	
Genetic algorithms applied to the scheduling of the Hubble Space Telescope	N89-26607		FIESTA: An operational decision aid for space network fault isolation	N89-10075
The 1990 Goddard Conference on Space Applications of Artificial Intelligence	N90-22294	SIGNS AND SYMPTOMS	Anomalous event diagnosis for environmental satellite systems	N93-25971
(NASA-CP-3068)				
The application of connectionism to query planning/scheduling in intelligent user interfaces	N90-22295	SIMULATION	An adaptive control scheme for a flexible manipulator	N89-10095
The role of artificial intelligence techniques in scheduling systems	N90-22297			
PLAN-IT-2: The next generation planning and scheduling tool	N90-22298	SIMULATORS	A university teaching simulation facility	N89-10088
An approach to rescheduling activities based on determination of priority and disruptivity	N90-22299			
Space communications scheduler: A rule-based approach to adaptive deadline scheduling	N90-22300	SOFTWARE ENGINEERING	The need for a comprehensive expert system development methodology	N88-30362
SURE (Science User Resource Expert): A science planning and scheduling assistant for a resource based environment	N90-22302		A hierarchical distributed control model for coordinating intelligent systems	N91-22783
Managing temporal relations	N90-22304		Knowledge-based approach for generating target system specifications from a domain model	N92-23371
System control of an autonomous planetary mobile spacecraft	N90-22313		Model-based reasoning for system and software engineering: The Knowledge From Pictures (KFP) environment	N93-25978
Optimum-AIV: A planning and scheduling system for spacecraft AIV	N91-22770		In search of meta-knowledge	N93-25983
Sharing intelligence: Decision-making interactions between users and software in MAESTRO	N91-22772		C++ Planning and Resource Reasoning (PARR) shell	N94-35054
Transformation reborn: A new generation expert system for planning HST operations	N91-22773		Engineering large-scale agent-based systems with consensus	N94-35071
Using C to build a satellite scheduling expert system: Examples from the Explorer platform planning system	N91-22774		Proceedings of 1986 Conference on Artificial Intelligence Applications	N90-70235
Long range science scheduling for the Hubble Space Telescope	N91-22775			
AI techniques for a space application scheduling problem	N91-22776		An expert system prototype for aiding in the development of software functional requirements for NASA Goddard's command management system: A case study and lessons learned	N90-70240
SLS-PLAN-JT: A knowledge-based blackboard scheduling system for Spacelab life sciences missions	N92-23358	SOFTWARE REUSE		
Detecting opportunities for parallel observations on the Hubble Space Telescope	N92-23359		The proposal entry processor: Telescience applications for Hubble Space Telescope science operations	N88-30339
Using AI/expert system technology to automate planning and replanning for the HST servicing missions	N93-25962		PST and PARR: Plan specification tools and a planning and resource reasoning shell for use in satellite mission planning	N89-26587
On-board emergent scheduling of autonomous spacecraft payload operations	N94-35047		Resource Allocation Planning Helper (RALPH): Lessons learned	N90-22296
A criterion autoscheduler for long range planning	N94-35048		Using C to build a satellite scheduling expert system: Examples from the Explorer platform planning system	N91-22774
Modeling heterogeneous processor scheduling for real time systems	N94-35053		Knowledge-based approach for generating target system specifications from a domain model	N92-23371
SCIENTIFIC VISUALIZATION		SOLAR RADIATION		
Interactive archives of scientific data	N94-35044		A statistical inference approach for the retrieval of the atmospheric ozone profile from simulated satellite measurements of solar backscattered ultraviolet radiation	N94-35060
RAVE: Rapid Visualization Environment	N94-35046			
SEARCH PROFILES		SPACE COMMUNICATION	Space Communications Artificial Intelligence for Link Evaluation Terminal (SCAILET)	N91-22796
Space communications scheduler: A rule-based approach to adaptive deadline scheduling	N90-22300			
SEARCHING		SPACE EXPLORATION	PLAN-IT: Scheduling assistant for solar system exploration	N89-10068
Expert systems built by the Expert: An evaluation of OPS5	N89-10086			
SEMANTICS		SPACE FLIGHT TRAINING	An intelligent training system for space shuttle flight controllers	N88-30331
Using expert systems to implement a semantic data model of a large mass storage system	N90-22315			
Visualizing the semantic content of large text databases using text maps	N93-25982			

SPACE MAINTENANCE	System control of an autonomous planetary mobile spacecraft
A situated reasoning architecture for space-based repair and replace tasks	N90-22313
SPACE MISSIONS	Automated procedure execution for space vehicle autonomous control
PLAN-IT: Scheduling assistant for solar system exploration	N90-22314
The achievement of spacecraft autonomy through the thematic application of multiple cooperating intelligent agents	N92-23364
An architecture for object-oriented intelligent control of power systems in space	N93-25968
Vista goes online: Decision-analytic systems for real-time decision-making in mission control	N94-35063
SPACE SHUTTLE MAIN ENGINE	The Generic Spacecraft Analyst Assistant (GenSAA): A tool for automating spacecraft monitoring with expert systems
Automatic mathematical modeling for real time simulation system	N91-22779
Detection of bearing failure in mechanical devices using neural networks	Expert operator's associate: A knowledge based system for spacecraft control
SPACE SHUTTLE MISSIONS	Distributed expert systems for ground and space applications
Mission Evaluation Room Intelligent Diagnostic and Analysis System (MIDAS)	N92-23361
SPACE SHUTTLE PAYLOADS	An autonomous satellite architecture integrating deliberative reasoning and behavioural intelligence
Spacelab data processing facility (SLDPF) Quality Assurance (QA)/Data Accounting (DA) expert systems: Transition from prototypes to operational systems	N93-25970
SPACE SHUTTLES	On-board emergent scheduling of autonomous spacecraft payload operations
Multi-viewpoint clustering analysis	N94-35047
SPACE STATION FREEDOM	A linguistic geometry for space applications
A failure diagnosis and impact assessment prototype for Space Station Freedom	N94-35050
A failure recovery planning prototype for Space Station Freedom	Vista goes online: Decision-analytic systems for real-time decision-making in mission control
Evaluating model accuracy for model-based reasoning	N94-35063
SPACE STATION PAYLOADS	Space/ground systems as cooperating agents
Maintaining consistency between planning hierarchies: Techniques and applications	N94-35072
An approach to knowledge engineering to support knowledge-based simulation of payload ground processing at the Kennedy Space Center	The design and application of a Transportable Inference Engine (TIE)
SPACE STATION STRUCTURES	N90-70241
Space station Platform Management System (PMS) replanning using resource envelopes	SPACECRAFT DESIGN
N89-10071	Space/ground systems as cooperating agents
SPACE STATIONS	SPACECRAFT EQUIPMENT
A shared-world conceptual model for integrating space station life sciences telescience operations	A method for interactive satellite failure diagnosis: Towards a connectionist solution
N88-30333	N89-26590
Integrated resource scheduling in a distributed environment	SPACECRAFT INSTRUMENTS
N88-30342	Capturing flight system test engineering expertise: Lessons learned
The space station assembly phase: System design trade-offs for the flight telerobotic servicer	N91-22790
AUTOPLAN: A PC-based automated mission planning tool!	SPACECRAFT LAUNCHING
N89-10067	Inferring heuristic classification hierarchies from natural language input
Autonomous power expert system	N93-25979
SPACEBORNE EXPERIMENTS	SPACECRAFT MANEUVERS
AI techniques for a space application scheduling problem	An expert system that performs a satellite station keeping maneuver
N91-22776	N89-10069
The use of artificial intelligence techniques to improve the multiple payload integration process	SPACECRAFT PERFORMANCE
N92-23370	Tracking and data relay satellite fault isolation and correction using PACES: Power and attitude control expert system
SPACECRAFT COMMUNICATION	N89-26592
A rule-based systems approach to spacecraft communications configuration optimization	Automated procedure execution for space vehicle autonomous control
N88-30341	N90-22314
CLEAR: Communications Link Expert Assistance Resource	SPACECRAFT POWER SUPPLIES
N89-10072	Tracking and data relay satellite fault isolation and correction using PACES: Power and attitude control expert system
SHARP: A multi-mission AI system for spacecraft telemetry monitoring and diagnosis	N89-26592
N89-26594	Autonomous power system intelligent diagnosis and control
Expert system development methodology and the transition from prototyping to operations: FIESTA, a case study	N91-22781
N89-26605	SPACECRAFT REENTRY
SPACECRAFT CONFIGURATIONS	Multi-viewpoint clustering analysis
Analyzing spacecraft configurations through specialization and default reasoning	N93-25980
N90-22308	SPACELAB
SPACECRAFT CONTROL	Spacelab data processing facility (SLDPF) Quality Assurance (QA)/Data Accounting (DA) expert systems: Transition from prototypes to operational systems
Automated space vehicle control for rendezvous proximity operations	N88-30353
Spacecraft command verification: The AI solution	Spacelab Data Processing Facility (SLDPF) quality assurance expert systems development
N90-22307	N89-10079
	Spacelab Life Sciences-1 electrical diagnostic expert system
	N89-26593
SPATIAL DISTRIBUTION	SPECTRAL DISTRIBUTION
Development of an intelligent interface for adding spatial objects to a knowledge-based geographic information system	N89-26598
SPECIFICATIONS	SPECTRAL REFLECTANCE
Knowledge-based approach for generating target system specifications from a domain model	VEG: An intelligent workbench for analysing spectral reflectance data
N92-23371	N94-35059
SPECTRAL SIGNATURES	SPECTRAL SIGNATURES
Improved image classification with neural networks by fusing multispectral signatures with topological data	N92-23368
STATIONKEEPING	STATIONKEEPING
An expert system that performs a satellite station keeping maneuver	N89-10069
STATISTICAL ANALYSIS	STATISTICAL ANALYSIS
Data exploration systems for databases	N92-23374
Automated analysis of complex data	N94-35056
A statistical inference approach for the retrieval of the atmospheric	

ozone profile from simulated satellite measurements of solar backscattered ultraviolet radiation	N94-35060	between TDRSS and ERBS	N89-10065
STOCHASTIC PROCESSES		An expert system that performs a satellite station keeping maneuver	N89-10069
An approach to rescheduling activities based on determination of priority and disruptivity	N90-22299	Tracking and data relay satellite fault isolation and correction using	N89-10069
STRATEGY		PACES: Power and attitude control expert system	N89-26592
The advice taker/inquirer: A system for high-level acquisition of expert knowledge	N88-30359	TDRSS momentum unload planning	N91-22771
STRUCTURAL DESIGN		TECHNOLOGICAL FORECASTING	
The space station assembly phase: System design trade-offs for the flight telerobotic servicer	N88-30357	Using AI/expert system technology to automate planning and replanning for the HST servicing missions	N93-25962
Kinematic study of flight telerobotic servicer configuration issues	N89-10100	TECHNOLOGY ASSESSMENT	
Actuators for a space manipulator	N89-10101	Actuators for a space manipulator	N89-10101
STUDENTS		TECHNOLOGY UTILIZATION	
Lessons learned in the development of the STOL intelligent tutoring system	N91-22795	The 1993 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3200)	N93-25961
SUPERCONDUCTING DEVICES		The 1994 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3268)	N94-35043
Adaptive pattern recognition by mini-max neural networks as a part of an intelligent processor	N90-22317	TELECOMMUNICATION	
SURFACE TEMPERATURE		Anomalous event diagnosis for environmental satellite systems	N93-25971
Neural networks for atmospheric retrievals	N93-25975	Telecommunications issues of intelligent database management for ground processing systems in the EOS era	N94-35067
SURVEYS		TELEMETRY	
A reusable knowledge acquisition shell: KASH	N91-22789	Mission telemetry system monitor: A real-time knowledge-based system	N88-30345
SYMBOLIC PROGRAMMING		Automated cataloging and characterization of space-derived data	N88-30354
Advanced data management design for autonomous telerobotic systems in space using spaceborne symbolic processors	N89-10096	SHARP: A multi-mission AI system for spacecraft telemetry monitoring and diagnosis	N89-26594
SYNTAX		An architecture for the development of real-time fault diagnosis systems using model-based reasoning	N92-23363
Knowledge structure representation and automated updates in intelligent information management systems	N90-22316	Mission Evaluation Room Intelligent Diagnostic and Analysis System (MIDAS)	N94-35064
Transformation reborn: A new generation expert system for planning HST operations	N91-22773	A distributed computing model for telemetry data processing	N94-35069
Simple methods of exploiting the underlying structure of rule-based systems	N90-70244	ISTAR: Intelligent System for Telemetry Analysis in Real-time	N94-35070
SYSTEM FAILURES		TELEOPERATORS	
A method for interactive satellite failure diagnosis: Towards a connectionist solution	N89-26590	Low level image processing techniques using the pipeline image processing engine in the flight telerobotic servicer	N88-30346
NASA ground terminal communication equipment automated fault isolation expert systems	N90-22305	The space station assembly phase: System design trade-offs for the flight telerobotic servicer	N88-30357
A failure recovery planning prototype for Space Station Freedom	N91-22778	Space truss assembly using teleoperated manipulators	N89-10087
An intelligent control system for failure detection and controller reconfiguration	N94-35066	A university teaching simulation facility	N89-10088
SYSTEMS ENGINEERING		Open control/display system for a telerobotics work station	N89-10089
The 1991 Goddard Conference on Space Applications of Artificial Intelligence (NASA-CP-3110)	N91-22769	Consolidated fuel reprocessing program: The implications of force reflection for teleoperation in space	N89-10090
Autonomous power system intelligent diagnosis and control	N91-22781	Issues, concerns, and initial implementation results for space based telerobotic control	N89-10091
Capturing flight system test engineering expertise: Lessons learned	N91-22790	A shared position/force control methodology for teleoperation	N89-10092
Evaluating model accuracy for model-based reasoning	N92-23362	Advanced data management design for autonomous telerobotic systems in space using spaceborne symbolic processors	N89-10096
The achievement of spacecraft autonomy through the thematic application of multiple cooperating intelligent agents	N92-23364	NASREN: Standard reference model for telerobot control	N89-10099
DIOGENES: Expert system for extraction of data system requirements	N90-70245	Kinematic study of flight telerobotic servicer configuration issues	N89-10100
SYSTEMS INTEGRATION		Teleoperated position control of a PUMA robot	N89-10104
Optimum-AIV: A planning and scheduling system for spacecraft AIV	N91-22770	Parallel plan execution with self-processing networks	N89-26584
SYSTEMS MANAGEMENT		The JPL/KSC telerobotic inspection demonstration	N90-22312
Techniques and implementation of the embedded rule-based expert system using Ada	N91-22788	Machine vision based teleoperation aid	N91-22784
T		TELEROBOTICS	
TASKS		Low level image processing techniques using the pipeline image processing engine in the flight telerobotic servicer	N88-30346
Space truss assembly using teleoperated manipulators	N89-10087	The space station assembly phase: System design trade-offs for the flight telerobotic servicer	N88-30357
Dynamic task allocation for a man-machine symbiotic system	N89-10098	Open control/display system for a telerobotics work station	N89-10089
TDR SATELLITES		Issues, concerns, and initial implementation results for space based telerobotic control	N89-10091
An expert system for scheduling requests for communications Links		Advanced data management design for autonomous telerobotic	

systems in space using spaceborne symbolic processors N89-10096
 NASREN: Standard reference model for telerobot control N89-10099
 Kinematic study of flight telerobotic servicer configuration issues N89-10100
 The JPL/KSC telerobotic inspection demonstration N90-22312
TEXTS
 Visualizing the semantic content of large text databases using text maps N93-25982
TEXTURES
 Feature detection in satellite images using neural network technology N92-23366
THEMATIC MAPPERS (LANDSAT)
 Classifying multispectral data by neural networks N93-25976
THERMAL MAPPING
 Image analysis by integration of disparate information N93-25973
TOOLS
 Cable applications in robot compliant devices N89-10102
TOPOLOGY
 Improved image classification with neural networks by fusing multispectral signatures with topological data N92-23368
TORQUE
 TDRSS momentum unload planning N91-22771
TRACKING (POSITION)
 A connectionist model for dynamic control N89-26604
TRACKING STATIONS
 Distributed expert systems for ground and space applications N92-23361
TRAJECTORY CONTROL
 A linguistic geometry for space applications N94-35050
 A genetic technique for planning a control sequence to navigate the state space with a quasi-minimum-cost output trajectory for a non-linear multi-dimensional system N94-35052
 A prototype expert system in OPS5 for data error detection N90-70242
TRAJECTORY PLANNING
 A genetic technique for planning a control sequence to navigate the state space with a quasi-minimum-cost output trajectory for a non-linear multi-dimensional system N94-35052
TRANSFORMATIONS (MATHEMATICS)
 Knowledge repositories for multiple uses N91-22797
TRUSSES
 Artificial intelligence approach to planning the robotic assembly of large tetrahedral truss structures N92-23357

W

WARNING SYSTEMS
 A proven knowledge-based approach to prioritizing process information N91-22791
WATER TEMPERATURE
 Neural networks for atmospheric retrievals N93-25975
WORKSTATIONS
 Open control/display system for a telerobotics work station N89-10089

U

USER REQUIREMENTS
 Integrated resource scheduling in a distributed scheduling environment N88-30342
 CLEAR: Communications Link Expert Assistance Resource N89-10072
 Robot hands and extravehicular activity N89-10097
 Design of an intelligent information system for in-flight emergency assistance N91-22792
 Proceedings of 1986 Conference on Artificial Intelligence Applications N90-70235
 (NASA-TM-101873)

V

VEGETATION
 VEG: An intelligent workbench for analysing spectral reflectance data N94-35059
VISUAL PERCEPTION
 Machine vision based teleoperation aid N91-22784
VOYAGER 2 SPACECRAFT
 SHARP: A multi-mission AI system for spacecraft telemetry monitoring and diagnosis N89-26594

The NASA STI Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Office plays a key part in helping NASA maintain this important role.

The NASA STI Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Office is also NASA's institutional mechanism for disseminating the results of its research and development activities.

Specialized services that help round out the Office's diverse offerings include creating custom thesauri, translating material to or from 34 foreign languages, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Office, you can:

- **Phone** the NASA Access Help Desk at (301) 621-0390
- **Fax** your question to the NASA Access Help Desk at (301) 621-0134
- **E-mail** your question via the **Internet** to help@sti.nasa.gov
- **Write** to

NASA Access Help Desk
NASA Center for AeroSpace Information
800 Elkrige Landing Road
Linthicum Heights, MD 21090-2934



Call for Papers



Goddard Conference on Space Applications of Artificial Intelligence and Emerging Information Technologies

May 1996

NASA Goddard Space Flight Center
Greenbelt, Maryland

The Eleventh Annual Goddard Conference on Space Applications of Artificial Intelligence and Emerging Information Technologies will focus on AI and IT research and applications relevant to space systems, space operations, and space science. Topics will include, but are not limited to:

- ⇒ Neural networks, genetic algorithms & fuzzy systems
- ⇒ High-Performance Computing & Communications
- ⇒ Knowledge-based spacecraft command & control
- ⇒ Expert system management & methodologies
- ⇒ Trend analysis, fault isolation & diagnosis
- ⇒ Distributed knowledge-based systems
- ⇒ Fault-tolerant/autonomous systems
- ⇒ Intelligent database management
- ⇒ Robotics & telerobotics
- ⇒ Planning & scheduling
- ⇒ Knowledge acquisition
- ⇒ Image analysis

Original, unpublished papers are now being solicited for the conference. Abstracts should be 300–500 words in length, and must describe work with clear AI or IT content and applicability to space-related problems. Two copies of the abstract should be submitted by September 1, 1995 along with the author's name, affiliation, address and telephone number. Notification of tentative acceptance will be given by September 22, 1995. Papers should be no longer than 15 pages and must be submitted in camera-ready form for final acceptance by November 27, 1995.

Accepted papers will be presented formally or as poster presentations, which may include demonstrations. All accepted papers will be published in the Conference Proceedings as an official NASA document. There will be a Conference award for Best Paper.

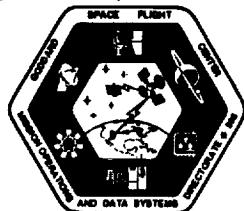
Please e-mail or FAX submissions if possible.

Visit our Conference Home Page at:

<http://defiant.gsfc.nasa.gov/aiconf/AI-conf-General.html>

No commercial presentations will be accepted

Sponsored by NASA/GSFC



Mission Operations and
Data Systems Directorate

1996 Goddard Conference on
Space Applications of Artificial Intelligence and
Emerging Information Technologies
May 1996 — NASA/GSFC, Greenbelt, MD

- Abstracts due:** Sept. 1, 1995 **Send abstracts to:**
 Papers due: Nov. 27, 1995 Walt Truszkowski
 Further info: (301) 286-3150 NASA/GSFC Code 522.3
 FAX: (301) 286-1768 Greenbelt, MD 20771
Walter.Truszkowski@gsfc.nasa.gov

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE May 1995	3. REPORT TYPE AND DATES COVERED Conference Publication - May 9-11, 1995	
4. TITLE AND SUBTITLE 1995 Goddard Conference of Space Applications of Artificial Intelligence and Emerging Information Technologies		5. FUNDING NUMBERS 510	
6. AUTHOR(S) Carl F. Hostetter, Editor			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS (ES) Goddard Space Flight Center Greenbelt, Maryland 20771		8. PERFORMING ORGANIZATION REPORT NUMBER 93B00058	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS (ES) National Aeronautics and Space Administration Washington, DC 20546-0001		10. SPONSORING / MONITORING AGENCY REPORT NUMBER NASA CP-3296	
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 63 Availability: NASA CASI (301) 621-0390.		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This publication comprises the papers presented at the 1995 Goddard Conference on Space Applications of Artificial Intelligence and Emerging Information Technologies held at the NASA/Goddard Space Flight Center, Greenbelt, Maryland, on May 9 - 11, 1995. The purpose of this annual conference is to provide a forum in which current research and development directed at space applications of artificial intelligence can be presented and discussed.			
14. SUBJECT TERMS Artificial Intelligence expert systems, planning, scheduling, fault diagnosis, control, knowledge representation, knowledge acquisition, neural networks, distributed systems, fuzzy logic, emerging information technologies			15. NUMBER OF PAGES 389
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL