NASA-TM-109733

/N -3 -TM

_ 0-11 J1

_/337

'/8 P

ARTIFICIAL INTELLIGENCE RESEARCH BRANCH

1992 Progress Report and Future Plans

PETER FRIEDLAND, BRANCH CHIEF
MONTE ZWEBEN, ASST. BRANCH CHIEF

AI RESEARCH BRANCH, MAIL STOP 269-2
NASA AMES RESEARCH CENTER
MOFFETT FIELD, CA 94025

EDITED BY HELEN STEWART

N/\S/\ Ames Research Center

Artificial Intelligence Research Branch
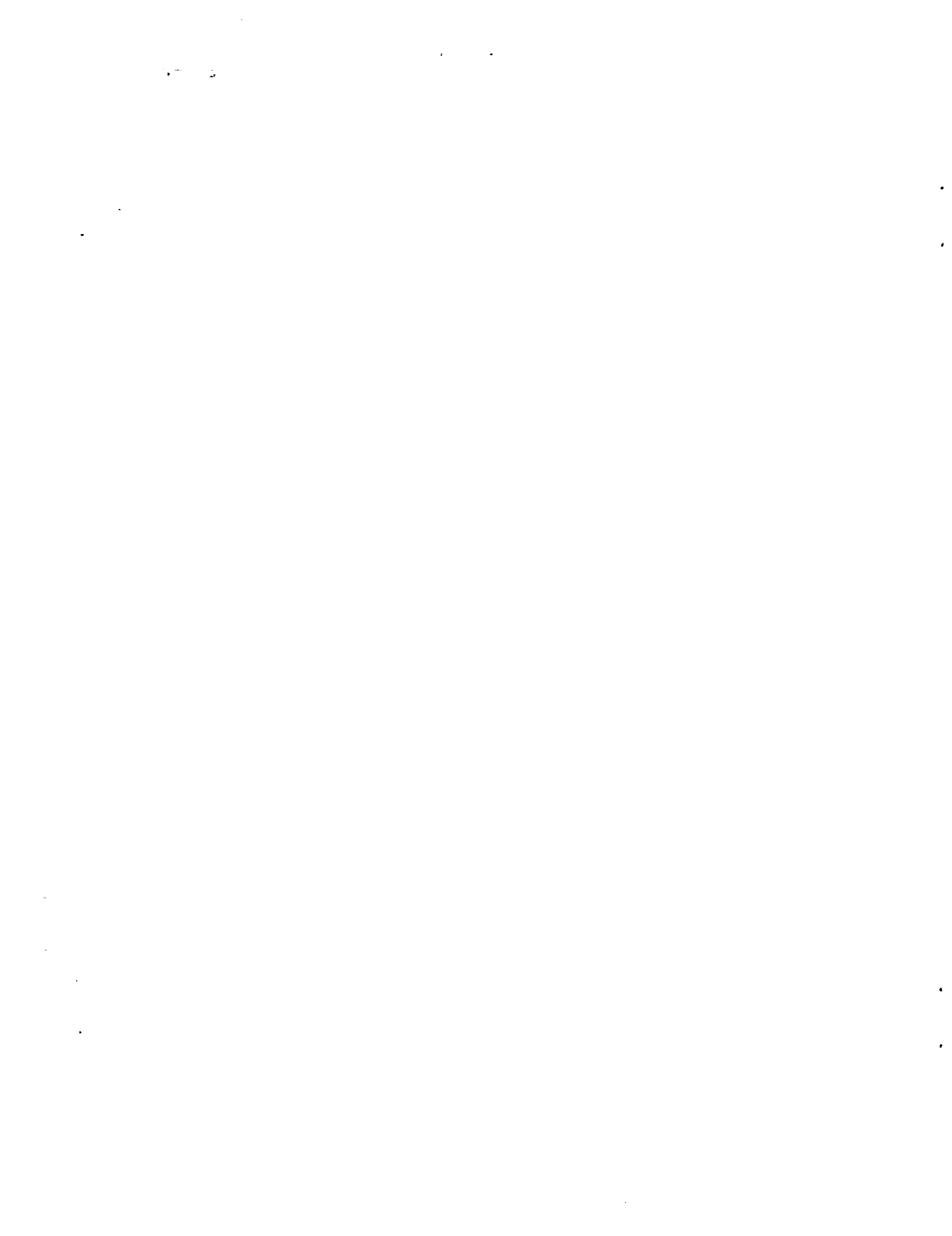
Table of Contents

INTRODUCTION

This report contains information on the activities of the Artificial Intelligence Research Branch (FIA) at NASA Ames Research Center (ARC) in 1992, as well as planned work in 1993. These activities span a range from basic scientific research through engineering development to fielded NASA applications, particularly those applications that are enabled by basic research carried out in FIA. Work is conducted in-house and through collaborative partners in academia and industry. All of our work has research themes with a dual commitment to technical excellence and applicability to NASA short, medium, and long-term problems. FIA acts as the Agency's lead organization for research aspects of artificial intelligence, working closely with a second research laboratory at the Jet Propulsion Laboratory (JPL) and AI applications groups throughout all NASA centers.

The year 1992 marked the maturation of the organization in two major respects. First, the FIA Branch now has approximately 60 full-time and visiting scientists and engineers with 22 holding a PhD in computer science or related fields and an additional 25 holding other advanced degrees. FIA staff participate actively in both the artificial intelligence and aerospace professional communities, serving as editors and editorial board members of several major journals, as members of the program committee of the National Conference on Artificial Intelligence, and as members of Technical Committees of the American Institute of Aeronautics and Astronautics (AIAA). Second, FIA moved into a state of the art new building, the Automation Sciences Research Facility. Besides the inter-group collaboration that physical proximity will enhance, FIA now has the laboratory space to conduct a wide variety of experiments on the integration of AI software with space hardware.

This report is organized along three major research themes:

- Planning and Scheduling: deciding on a sequence of actions to achieve a set of complex goals and determining when to execute those actions and how to allocate resources to carry them out.

- Machine Learning: techniques for forming theories about natural and man-made phenomena; and for improving the problem-solving performance of computational systems over time.

- Research on the acquisition, representation, and utilization of knowledge in support of diagnosis design of engineered systems and analysis of actual systems.

Our work is funded by a variety of sources from NASA and other federal agencies. The single largest sponsor of work is the Artificial Intelligence Program of the Information Science and Human Factors Division of the NASA Office of Aeronautics, Exploration, and Technology (OAET). (Dr. Mel Montemerlo is the manager of this program). We also receive support from the Space Station Freedom Advanced Development Program for the PI-in-a-Box Project and from the Office of Space Science and Applications Information Systems Division for the SIGMA Project. (Mr. Mark Gersh and Dr. Joe Bredekamp are the responsible program managers.) The Defense Advanced Research Projects Agency's Information Science and Technology Office (DARPA/ISTO) (Major Steve Cross is program manager) co-funds our work in Intelligent Interacting Agents.

PLANNING AND SCHEDULING

Space Shuttle Ground Processing Scheduling
Monte Zweben , Eugene Davis, Ellen Drascher, Brian Daun, Michael Hall
Michael Deale, Donna Kautz, Mark Yvanovich, Danielle Schnitzius, Michael Carpenter; Lockheed
Space Operations Company

The Space Shuttle Ground Processing Scheduling project is deploying a constraint-based scheduling system at the Kennedy Space Center (KSC). The goal of the project is to assist KSC management in streamlining Shuttle processing. The KSC flow management team has the responsibility for preparing the Space Shuttle for flight. Shuttle processing (referred to as the *flow*) actually begins while the orbiter is in space. The flow management team meets the orbiter at its landing site and performs a structural inspection of the orbiter with the hope of making an accurate assessment of any damage that might have occurred during the mission. Based on this early assessment, the flow management team modifies its ground processing schedule accordingly. The first activity of the flow is the orbiter's arrival at the KSC Orbiter Processing Facilities (OPF). There, each orbiter subsystem is inspected, repaired, and tested. The OPF consists of large hangars capable of housing three orbiters where volatile materials and large objects are manipulated. This process usually extends for about 65-90 days, after which the orbiter is transported to the Vehicle Assembly Building (VAB). There, the orbiter is mated with the Solid Rocket Boosters and the External Tank. Subsequently, the vehicle (i.e., orbiter, boosters, and tank) is transported to the Launch Pad for final launch preparations.

The processing within the OPF is comprised of a set of extremely dynamic and complex activities coordinated by the flow management. Execution of a ground processing schedule is plagued by the uncertainty inherent in most complex operations environments. For example, delays are caused by backordered parts, contention for resources, and competition for conflicting configurations of the orbiter. Additionally, equipment often malfunctions and personnel get sick. The uncertainty of this environment is heightened by the fact that it is impossible to completely plan the processing activities in advance because unpredictable problems are detected during inspection tasks. As a result, dynamic rescheduling is a crucial aspect of Space Shuttle processing.

The Ground Processing Scheduling System (SSGPS) develops schedules with respect to temporal, resource, and state constraints. The attributes of the "environment" that change over time are represented as state information. This includes the configuration of the orbiter as well as the spatial zones within the OPF. Orbiter configuration includes the positions of the landing gear, payload bay doors, elevons, rudder speed brake, and status of the orbiter power and hydraulics systems. SSGPS uses constraint-based iterative repair as its search method. The approach begins with an initial, flawed schedule and then iteratively repairs constraint violations until a conflict-free schedule is produced.

1992 Progress

Since SSGPS can model state constraints, state information had to be acquired for each task in the flow. To gather this information we developed a number of knowledge editors for the flow managers. Eric Clanton (Endeavour) and Wayne Bingham (Columbia) visited Ames for a week for an extensive period of knowledge engineering. These tools enabled the flow managers to quickly populate our knowledge base by designating temporal, resource, and configuration constraints for each task. In order for this information to be useful across multiple flows, we built a database mapping tool that allows the user to map a well-engineered data set onto one downloaded from the existing KSC scheduling system (CAPSS). This tool matches tasks in one dataset to analogous tasks in a new set and maps the resource and configuration tasks from the old set onto the new one. We are currently extending this tool so that temporal constraints could also be inherited from these engineered data sets.

During 1992, the STS-49 field study was completed. SSGPS was used to track the schedule of four months of the first Endeavour flow. This task required the KSC team to attend all Endeavour scheduling meetings and subsequently input any schedule changes into SSGPS. The goal of this experiment was to investigate the performance of the system under real-time constraints and to try to assist Eric Clanton, the flow manager of Endeavour, in streamlining the schedule. Our experiment was successful in that we were able to generally keep pace with the manual scheduling process at KSC, but we were not able to provide much insight into streamlining the schedule. Improving the Endeavour flow proved difficult because the maiden flight of an orbiter is quite unique and one has very little information as to what tasks will ultimately be necessary to meet the launch requirements. For instance, 2500 unexpected tasks have been added to the STS-49 flow, and the KSC team entered only a fraction of them into SSGPS. Moreover, the tasks entered into SSGPS did not contain orbiter configuration and resource requirements because it was not available in a central database or from a single person (as far as we knew). Therefore, the impact of these tasks was only reflected via the temporal constraints defined on them. The second cause of difficulty encountered during STS-49 was data integrity. Our data originates within the CAPSS planning and scheduling tool which is mostly used to support temporal networking. KSC does not use the resource modeling capability of CAPSS and CAPSS can not model orbiter configuration, thus the only way to constrain the placement of tasks on a timeline is through the use of temporal relations (such as predecessor-successor relationships). However, when tasks are dynamically rescheduled, these artificial predecessor-successor relationships (used to compensate for resource and configuration constraints) become obsolete and one must debug the network of constraints to produce acceptable schedules. To illustrate, suppose two tasks compete for the same piece of heavy equipment. The CAPSS team will constrain the tasks to have one follow the other with a predecessor-successor relationship. If the first task is delayed for some unexpected reason, then SSGPS will propagate the constraint and inappropriately move the second task later as well. Most of the time updating the daily STS-49 flow is spent debugging this network of constraints leaving little time for analysis of the schedule with respect to other constraints.

To combat this data integrity problem, we must sanitize any data set downloaded from CAPSS and eventually use our own library of tasks exclusively. The tools developed during the flow managers' visit to Ames will be enhanced for this purpose. While this experiment yielded mixed results, we did have one of our system's schedules distributed to the KSC community for work taking place during the Christmas break. This was the first official use of an SSGPS schedule.

The SSGPS graphical interface was translated to the MOTIF interface standard to enhance performance, to improve the aesthetic "look and feel", and to guarantee portability. This effort required substantial modification to the original interface. Additionally, a PostScript report generator was developed that allowed SSGPS to output paper schedules that are identical in style to those that are manually developed and accepted by the KSC operations community.

The SSGPS search mechanism was extensively enhanced with an optimization capability, an improved state constraint repair, and a focus mechanism. One of the driving costs of space shuttle processing is overtime labor. An optimization constraint was developed that improves schedules by changing their legal calendars. A calendar specifies the days of the week, shifts, and holidays that are legal for a task. Tasks are preempted and restarted according to these calendars. The optimization constraint changes the calendars on tasks from those that allow undesirable work periods, to those that avoid these periods. For example, some tasks are modified to avoid weekend work. However, this repair can extend the schedule and possibly cause a task to extend past a milestone. The optimization constraint monitors the search progress and retracts its changes if it appears that the new calendars prohibit the system from finding acceptable schedules.

Additionally, we have re-engineered the search mechanism with respect to state constraints. The search now avoids superfluous state transitions. For example, our system suggested schedules where the orbiter power would cycle on and off frequently, which would be unworkable in the OPF. Consequently, for some orbiter configurations, our search mechanism sets a state for an entire shift thus avoiding this problem. Additionally, we have added a simple planning component

to the repair of violated state constraints. In the previous implementation tasks with violated state constraints were moved to other times when the state constraint is satisfied. SSGPS now has the ability to create new tasks that achieve the correct state without moving the violated tasks. However, the system must track the dependencies of these supporting tasks because if all the tasks that were depending on them are moved (possibly to repair other constraints), then the added tasks are obsolete and must be removed.

A focus mechanism will also be implemented in 1992. This will allow the user to specify what constraints are relevant in the search process. This facility will have a graphical interface to support the flow managers in what-if analysis.

The highlight of 1992 is that SSGPS will be used operationally for STS-50 (Columbia). We plan on a 2-3 week shadow period where Wayne Bingham will use the manually developed schedules and then if the performance of SSGPS is adequate, it will be used for one month of the flow. During this month we will attempt to use the new focus and search mechanism to improve the schedule as well as provide the daily paper schedules. Our goal is to achieve quantifiable improvements that will be used to convince the projects organizations at KSC that SSGPS is a useful tool worthy of their support. If successful, these organizations could assist in the documentation and maintenance of the system and in the training of operations personnel.

1993 Plans

In 1992, we plan to fully document the system and release it through COSMIC for general distribution. We will continue to integrate our system with the existing operational databases and prepare the system for delivery to the KSC support organizations. With operational experience we will gather more knowledge from the flow managers and incorporate this into our search mechanism.

The flow managers have requested three main extensions to the existing system: an explanation facility, multiple orbiter schedules, and more detailed schedules. Multiple orbiter schedules are necessary because of critical resources that are shared across orbiters. For example, schedules are often delayed because one orbiter team might be using equipment that another team needs. If these assignments are not carefully coordinated, missions could miss their milestones because non-critical path tasks in another mission are using the scarce resources.

Flow mangers need to justify their decisions to KSC management and thus any schedule that SSGPS creates must be defended. To support this need, we will develop an explanation facility that allows the user to enter explanations for user-entered changes and that enables the search mechanism to justify its decisions.

The tasks that SSGPS presently schedules are gross procedures that have a great deal of detailed specializations. The flow managers would like the system to reason at a more detailed level for all the tasks that fall within a current 11 day window. To support this we plan to load the more detailed schedules for an 11 day window but leave the remaining schedule at the higher level abstraction. As the schedule is executed by the system it will revert the completed tasks to the higher level of abstraction or write them to disk to conserve memory. As the days advance, the system will automatically flesh out the detail of tasks entering the 11 day window.

Milestones

1992: - Shadowing of STS-49
- STS-50 Operational use of system by flow management assisted by LSOC personnel
- Optimization criteria added to system will improve overall flow.
1993: - Distributed access to scheduling data by KSC community
- System used by flow management autonomously.
- Multi-flow scheduling enabled.
- Mini-flow scheduling enabled.
1994: - System used routinely during scheduling meetings.

Iterative Improvement of Wind Tunnel Schedules:
The Wind Tunnel Scheduling Project
Kimball Collins, Gene Davis, Monte Zweben

The goal of this project is to work with the wind tunnel operations branches at the Ames Research Center (ARC) to develop scheduling software for predicting and reducing both the costs and energy consumption of ARC wind tunnel tests. Specifically, we are employing a constraint-based iterative improvement strategy in this hierarchical, fixed-preemptive scheduling domain.

This domain is hierarchical in that aerodynamic tests consist of multiple test model configurations. A series of runs are performed for each configuration, during which, normally, the wind tunnel blower operates continuously. Data points are in turn gathered from time to time during a run while model and environmental parameters are varied. Model parameters include, for example, pitch and yaw, while environmental parameters may include such factors as air speed and pressure.

The wind tunnel domain is also a fixed-preemptive one in that tasks are performed according to a changing work calendar that specifies electrical cost rate periods, labor schedules and availability, and whether weekend or other shift operation can take place. Tasks may sometimes be split among different rate and time periods and are associated with a work duration that remains constant when their actual duration changes. One aspect of these calendars is that they vary throughout the year, and change according to budgets and even cost predictions themselves.

This project is collaborative with the ground processing scheduling project (the KSC project). The base scheduling and interface software for the KSC project, GERRY, is being generalized and extended to handle this new domain and type of schedule.

The wind tunnel scheduling effort has two thrusts. First, modelling of the wind tunnel domain must be accurate in order to enable adequate prediction of power usage. Knowledge of wind tunnel operations and scheduling in the aerodynamic facilities branch is distributed over approximately three levels of detail. Human schedulers are responsible for high level decisions for when wind tunnels may run and what wind tunnel limits may be set. In addition, they produce estimates for various components of the fiscal year power bill. ARC test engineers work closely with aerodynamic engineers to determine configuration and run schedules for optimal experimentation. Finally, many wind tunnel mechanics and operators have specialized knowledge of safe operating limits, idiosyncrasies, and other aspects of operation. In 1991 we collected and assimilated this knowledge to formulate a model of operations that can be turned into a platform for implementing schedule improvement strategies.

The second thrust of this effort is the development of search strategies and heuristic repair methods to improve schedule cost and quality. Schedule cost can be determined by taking into account rate schedules, predicted power consumption, labor cost, and similar criteria. Schedule quality is more difficult to quantify. This includes such factors as minimal schedule perturbation during rescheduling, minimal test length, fairness among test customers and wind tunnels, etc. These tradeoffs must be accounted for and made adjustable by the end user. In 1991 we created a means of determining schedule cost and normalizing it to a 0-to-1 scale in such a way that as costs rise, the metric rises monotonically and has a variable-slope steep section placeable where the most critical region for cost improvement is deemed likely. The usefulness of these parameters will be tested in 1992 on the simulated annealing algorithm developed for the KSC project. In 1992 we began splitting prediction capability into two main components: test-level and run-level.

Test level scheduling is for coarse-grain prediction where only a few parameters are available for predicting the next fiscal year's costs. This is necessarily an error-prone process and the automation of it is intended to enable scenario playing and assist in determining likely maximum and minimum costs. Test level scheduling is over a period of several months or a year, and involves only the information that is available to the center approximately one year in advance.

Run level scheduling has approximate configuration schedules as input and is intended to achieve cost reduction through several means. Tasks can be optimized with respect to rate calendars, to likely wind tunnel maintenance needs, to other wind tunnels, and to optimal ordering. An example of optimal ordering is to arrange runs so that they range from slow runs to fast runs, so that the blower motor is monotonically increasing. The general problem is NP-complete, and the KSC project's iterative improvement approach is used for both their anytime characteristic and the ease with which user interactivity can be facilitated.

In 1991, the fixed-preemptive scheduling capability of the GERRY scheduling system was extended to accommodate periodic calendar changes. Future changes are to tie calendars to tests in such a way that weekends and shifts are dynamically able to be made valid or invalid for task placement.

The system, when it is fully functional, will be used by test engineers, human schedulers, and possibly technicians. It will be distributed over several wind tunnels and across two ARC branches, each with its own operating procedures, conventions, configuration, power use, and other requirements. While the various operators of the wind tunnels are all dedicated to the goal of performing the best tests for the least cost, communication is a useful contribution of a scheduling system, as each operator can potentially see the effect that schedule changes within his purview have on overall cost.

Future work in machine learning will be investigated in two ways. First, currently the power usage of wind tunnel runs is estimated manually, using intuitive techniques and graphs drawn from average data. In 1991 and 1992 we initiated a spin-off project with Wray Buntine and Richard Kraft to apply regression techniques for inducing a general formula to predict power usage. At the run level this prediction makes use of multi-variable data available from past runs; at the test level, discrete and more approximate variables will come into play. We anticipate that the test level estimates will be able to incorporate the formulas that we are inducing from the run level data.

The other way in which current research in machine learning has promise is in learning macro operators, discovering bottlenecks in a schedule, and developing better heuristics and schedule repair mechanisms. This work will proceed as a sideline to the main thrusts.

Milestones:

1992: - Completion of test level modelling and knowledge engineering.
       - Operation on line by human schedulers for cost prediction and scheduling.

GEMPLAN/COLLAGE Multiagent Planning Systems
Amy Lansky, Andrew Philpot

This work focuses on the problem of generating plans for domains with multiple streams of activity that require complex forms of coordination. Thus, it deals with both action generation and action ordering, and must handle such issues as resource allocation and scheduling. GEMPLAN and its successor system, COLLAGE, are unique in that they provide a broader range of plan construction techniques than most planners, in their use of localized search (which partitions a global search space into smaller, regional search spaces) to control the inherently explosive cost of planning, in their integration of pre-planning and dynamic application of constraints, and in their ability to flexibly tune search control.

1992 Progress

Work in 1992 focused in several areas. The most important work was in the design and development of the new COLLAGE system. COLLAGE now replicates or exceeds much of GEMPLAN's functionality, and successfully runs the GEMPLAN house-construction domain. In construction domains, COLLAGE takes the role of a facility general contractor, constructing a plan

of action that results in a finished artifice and that takes into consideration all the requirements and constraints of the building, subcontractors, and resources. We have run the same timing experiments in COLLAGE as performed in GEMPLAN (testing various domain localization topologies and tradeoffs), and the results confirm our prior analytical results and empirical experiences with GEMPLAN.

COLLAGE's constraint application and search facility has been designed to achieve much greater generality than GEMPLAN's. The system is highly object-oriented and extensible, constraint satisfaction algorithms are fully backtrackable over every choice point within an algorithm (thus achieving a degree of search completeness not available in GEMPLAN), and plan and search structures are extensively cross-referenced to allow for efficient access. The constraint algorithms now implemented in COLLAGE include all those in GEMPLAN except for the STRIPS algorithms and regular-expression pattern algorithms. COLLAGE's CSP constraint facility is much more general than GEMPLAN's, allowing for arbitrary functional constraints on plan variable values, and thus enables metric-time and other resource constraints. COLLAGE's nonatomic event decomposition strategy is also more general than GEMPLAN's.

We have also done a fair amount of design work towards future development of COLLAGE, especially more flexible control over search. In the earlier part of the year, we reimplemented and optimized GEMPLAN's search facility to allow for a broader range of localization structures. This was needed in order to allow for future experimentation with automatic generation of localizations (see below). These experiences with GEMPLAN enabled us to create a more efficient and correct design of COLLAGE's search facility. Although COLLAGE now utilizes GEMPLAN's fixed search strategies for maintaining global plan consistency and for constraint activation, we are now developing new approaches that are much more flexible and integrated. The basic idea will be to propagate plan changes as they occur, both to maintain consistency and to trigger affected constraints throughout the network of regional search trees. After constraints have been activated, meta-level control strategies (specializable on a per-region basis) control both constraint activation within a region as well as shifting between regional search trees. This basic mechanism for constraint activation is somewhat like that of a truth-maintenance structure, but less semantically deep and expensive. It will also be used as a framework for several other new developments in COLLAGE: run-time activation of constraints; encoding of justification structures to enable plan reuse and replanning; integration of loops that are "unraveled" at run-time.

Another focus of 1992 was on developing an application domain in the area of data analysis. Along with Rich Keller, we completed a revised version of a whitepaper that describes our "Data Analysis Workbench" concept. We then proceeded to identify specific problems that our respective systems could address that would be useful for earth scientists. After many meetings with NASA Ames ecologists, we decided to work on disjoint but complementary aspects of data analysis. Keller's system is being applied to modeling of the Forest BGC ecosystem model. COLLAGE will be used as a general purpose data registration planner to provide registered data input to models such as the one being constructed by Keller. It is estimated that greater than 50% of the scientist's time is spent on data preparation issues. We learned about the data registration process, including such topics as geometric correction, projection, scaling, and ground point determination. Along with Jennifer Dungan, one of the Ames ecologists, we then designed a general purpose algorithm for registration. We are now in the process of encoding this algorithm as domain input to COLLAGE.

The registration problem will exercise several new abilities in COLLAGE, including integration of user input into the planning process. This will be done by incorporating special types of actions and constraint forms that allow for user interaction - something more readily attained in the COLLAGE framework than in many planners. User input is essential for the EOS domain; only the scientist-user will know the full data registration requirements, and most scientists will want to have final say on the form of a generated plan. Indeed, for large coordinative domains like those targeted by COLLAGE, user control will always be important - a building contractor or human scheduler, for instance, would never wish to yield control completely to an automated system. The registration domain also has significant need for dynamic replanning, since faulty data often

results due to incorrect registration or modeling assumptions. As a result, scientists are constantly re-registering data for input into their models.

Finally, another area of research has been on automatically generating good domain localizations. Although we did fairly extensive work in GEMPLAN to prepare for this experiment, we have now decided to switch efforts completely to COLLAGE and perform this work in the new system. However, we will be able to reuse the ideas already developed for GEMPLAN, including algorithms for generating a first-pass localization and for localization transformations. We will also be performing various experiments in COLLAGE that test the interaction between localization strategies and such factors as: degree of backtracking, size of regional plans, and constraint difficulty. We will be conducting our localization experiments in a modified, parameterized version of the house construction domain, as well as an office-building construction domain. We anticipate that our work with construction domains will have potential in assisting with NASA's infrastructural requirements and will also have application to space construction. Towards this end, we have had ongoing contact with Steve Jolly of the Center for Space Construction in Colorado.

1993 Plans

Our future plans for 1993 involve continuing along the research paths already described above:

(1) Continued development of COLLAGE. This will include work on search control, dynamic constraint activation, addition of new constraint forms, and integration of user-input into the planning process.

(2) Development of the EOS data-registration planning domain and scientific workbench prototype.

(3) Experimentation with different localization strategies, and the development of automated localization techniques. Construction domains will be the primary testbed for these experiments.

In addition , we see larger efforts in the area of facilitating scientific data analysis. We believe that the data analysis workbench concept developed by Keller and Lansky could be fruitful in leveraging the abilities of scientists to make use of EOS data. We are developing our ideas in a bottom-up fashion, applying our modeling and planning technologies to specific subproblems of the data analysis tasks. Ultimately, however, the full workbench concept could form a very large project that integrates modeling and planning capabilities, as well as data visualization, data manipulation techniques (e.g., use of Autoclass to classify phenomena), and access to the EOSDIS distributed data base.

Milestones:

1992: - Complete an initial demonstration of the registration planning domain in COLLAGE.
- Experimentation with various localization strategies on a parameterized construction domain.
- Complete a first cut at a localization generation algorithm.
1993: - Integration of flexible search control strategies into COLLAGE, including meta-level control, run-time constraint activation, and user-input into the planning process. -
- Continued integration of new constraint forms.
- Publication of work on localization experiments and automatic generation of localization.
- Further development of the registration planning and construction planning domains.
- Development of a rudimentary run-time environment and user interface.
1994: - Development of a COLLAGE user interface that fosters a sophisticated level of user interaction.
- Further development of the EOS workbench architecture concept.
1995: - Experimentation with distribution of the COLLAGE search trees over multiple machines and a focus on distributed reasoning in general.
1996: - Experimentation with learning in COLLAGE.

Planning, Scheduling, and Control: An Integrated Approach

Mark Drummond, John Bresina, Keith Swanson, Rich Levinson, Andy Philips
Bill Borucki, Ames Theoretical Studies Branch

The overall goal of this project is an understanding of integrated planning, scheduling, and control. Work in the project is attacking the problem from the perspective of both theory and practice. In terms of theory, we are working to understand the basic principles by which intelligent planning, scheduling, and control systems may be designed, implemented, and analyzed. In terms of practice, we are working on the construction of a planning, scheduling, and control system for the management of photoelectric telescopes. We are actively working with NASA astronomers to design our system. Our goal is to build a useful, fully-functional planning, scheduling, and control system by the end of 1993.

1992 Progress

We have made significant progress in developing our basic theory. First, we have completed implementation of a discrete event simulator called The NASA TileWorld (NTW). This simulator is described by two technical reports. The first report presents an NTW design rationale, and the second one contains a detailed users' manual, complete with programmers' notes. The NTW simulator has been used extensively within our research project, and has also been distributed to many other sites throughout the country. The NTW has been submitted to COSMIC for ease of future distribution.

The NTW simulator has allowed us to define and carry out a detailed study of the value of planning ahead. The results from this study show that it is possible to define a reactive system and planner combination that exhibits the best qualities of the two systems in isolation. Specifically, it is possible for the reactor to act in bounded time, and the planner incrementally, monotonically, improves the reactor's performance until, in the limit, the reactor's performance is optimal. A detailed technical report describing these results has been written, and a journal paper is planned.

Other theoretical work involved a study of the costs and benefits of planning with total and partial orders. This work, while largely theoretical, did involve an interesting empirical component. One result from this work clearly demonstrates that the use of heuristic search control knowledge can easily be more important than the use of partial orders. A detailed technical report describing this and other results has been written. This work is collaborative with Steve Minton.

We have also had preliminary success with a new approach to problem-solving that we call "behavior-based search." The basic idea is to provide the search engine with the reactor's basic program, and for the search engine to use that program to guide its search. This idea has been studied to some degree, but more work is clearly warranted.

An Automatic Photoelectric Telescope (or APT) is a telescope controlled by a dedicated computer for the purpose of gathering photometric data about various objects in the sky. A "group" is the primitive unit to be scheduled and consists of a sequence of telescope and photometer commands defined by an astronomer. Any given astronomer has certain scientific goals, and he or she uses the group as the primary unit of instruction to an APT in order to achieve those goals. The language used to define groups is called ATIS (for Automatic Telescope Instruction Set); ATIS is the *de facto* standard language for communicating with APTs. Various astronomers define their requests in ATIS and these requests are submitted to a remotely located telescope for execution. While the current generation of telescope software can execute these ATIS files without human intervention, there is currently no look-ahead scheduling performed. The result of our work will be a look-ahead scheduler for ATIS request files that can make more efficient use of scarce telescope resources.

We have also made excellent progress towards demonstrating these basic research ideas on the problem of controlling automatic photoelectric telescopes. To date, we have built a special "editor" that allows a telescope manager to view and manipulate scientists' observation request

files. This system has met with great success, and has already been used by our space science collaborators to improve existing telescope request files. We have acquired a telescope simulator, and are now in a position to actually purchase an automatic telescope suitable for evaluating our prototype system. A technical report has been written that describes the telescope problem and it explains how our theory provides a solution. We also co-authored a book chapter that shows how a system such as ours can help reduce the operational costs of proposed lunar telescopes. This work has been presented at a number of different astronomy workshops and it has met with enthusiastic support each time. We have received a number of offers from astronomers who would like to test our system as it develops.

1993 Plans

Our theoretical objectives for next year include studying behavior-based search in more detail; specifically, we plan to define and execute automatic telescope experiments designed to discover its strengths and weaknesses. We also plan to further develop our nascent theory of "dynamic dependencies". This theory, when better developed, will allow our planner/scheduler to track relevant conditions in the environment and to control search based on the progression of external events. We also plan to explore the use of our (existing) algorithms for managing uncertain actions in the context of pure scheduling problems. In so doing, we expect to develop an explicit model of the tradeoff between schedule "optimality" and schedule "robustness." Finally, we plan to study the use of machine learning techniques in the area of composing reactive strategies provided by a human programmer. All of these theoretical topics have a potential impact on the telescope control problem, and we propose to study connections as they are established.

In terms of the telescope control problem itself, we first plan to get our existing "temporal projection" algorithms to work with our existing request file editor. We then plan on studying the compilation of situated control rules, as described in our theoretical work of two years previous. Once this functionality exists, we propose to change the existing telescope controller software to better interface with our external planner/scheduler system. Following this, we plan on testing our system against the telescope simulator in order to better develop metrics by which to judge system performance. There is also an excellent opportunity for "sensory planning" in the telescope control problem, and we plan to better define and study it.

Milestones

1992: - Define and study ``behavior-based" search.
　　　　 - Extend theory of dynamic dependencies.
　　　　 - Study the tradeoff between schedule "optimality" and "robustness".
　　　　 - Study learning techniques for composing reactive strategies.
　　　　 - Implement temporal projection for telescope control.
　　　　 - Implement situated control rule compilation for telescope control.
　　　　 - Modify existing telescope controller to interface with our system.
　　　　 - Establish metrics to judge system performance.
　　　　 - Study telescope ``sensory planning" problem.
1993: - Design and implement telescope versions of previous experiments.
　　　　 - Make telescope available to the AI community (for benchmarking purposes).
　　　　 - Document existing telescope control problem and architecture.
　　　　 - Extend end-user capabilities of telescope system.
1994: - Study other sorts of instruments, and how they impact telescope planning.
　　　　 - Work with astronomers to make our planning, scheduling, and control system widely available.

Computer Integrated Documentation

Nathalie Mathé, Bharathi Raghavan, Joshua Rabinowitz, Yaron Gold
Guy Boy, ONERA, France, Sterling Software Consultant

The Computer Integrated Documentation (CID) project integrates hypermedia and knowledge-based systems capabilities. The goal of the CID project is to facilitate intelligent interaction with humans in computerized information indexing and retrieval tasks. A CID workstation is being developed to help Space Station Level I personnel manage the Program Requirement Document for the Space Station Freedom and to give astronauts and ground personnel easy access to relevant information in both normal and abnormal operating conditions. Preliminary results with this first prototype has shown the value of context-dependent indexing and information retrieval. Moreover, this system is able to acquire knowledge continuously during its interaction with users, improving the smoothness of its performance over time.

Hypertext is a good tool for the development of documentation systems. Hypertext systems increase accessibility, but they do not provide any built-in selectivity mechanism. In other words, while non-linear or hypertext systems may dramatically increase the accessibility of information, this increased accessibility may magnify an already severe problem of selection. For these reasons, knowledge-based systems technology can be very helpful in alleviating the selection problem and the cognitive overhead of the user. Our basic approach on the CID project is to develop an intelligent context-sensitive browsing tool which interacts with and learns from the user, and uses advanced interaction media such as hypertext, multimedia, and virtual environments. Context-sensitive information retrieval provides the user with the information that is likely to be relevant to him/her in context. This is very helpful when navigating in large document systems and when there are many different users.

1992 Progress

The CID project has concentrated on four major activities: (1) the development of a generic hypermedia system augmented by an intelligent indexing and information retrieval mechanism; (2) the study of incremental context acquisition in information retrieval; (3) the application of machine learning to context-sensitive indexing; and (4) the application and first evaluation of the CID tool on the Program Requirement Document for the Space Station Freedom (SSF).

The context-sensitive information indexing and retrieval methodology is based on a knowledge representation (KR), called "Block KR", that we previously developed as a representation framework for operation manuals, procedures checklists, and other on-line tools useful for controlling and repairing complex dynamic systems. A knowledge block includes five components: contextual conditions, triggering preconditions, a set of actions and attached abnormal conditions, and a goal. The Block KR supports a learning mechanism that is based on acquisition of new blocks and refinement of existing blocks by adding or reinforcing abnormal conditions and contextual conditions. This representation has been efficiently implemented in the CID system to represent links between descriptors and referents. To automatically help the user, the problem is to "contextify" the links between documentation nodes, that is, to provide relations between descriptors and referents that are valid in the current context. These relations vary depending on the situation and the user. This context reduces the number of possible referents for a descriptor that a user has to consider, and thus speeds his or her search.

An algorithm called IARC (Index Acquisition and Refinement according to Context) has been developed. We define the context acquisition problem as discovery of "abnormal conditions" and generation of recovery actions, as well as reinforcement of current actions. Our system observes the user's actions during the browsing tasks, and, by noting whether a specific referent was considered a success or failure by the user in a particular context, it is able to refine the indices to reflect their context of use. In this way, our system automatically acquires the knowledge necessary to operationalize a user model: which indices are appropriate when, and for which user.

The current system has been tested on a small but significant subset of SSF documentation (a Technical Memorandum is available describing these results in detail). The CID system has also been successfully used for the F-18 emergency procedures management and maintenance. Besides providing an intelligent interface for browsing large documents, the ability of CID to acquire automatically the context in which strategies are appropriate is significant. First, it allows the system to provide a *tailorable* browsing facility. Indeed, CID will learn which referents are to be presented for which user. Second, it shows that it is feasible to immediately incorporate the user's feedback into the CID's knowledge, with the possibility of thus improving the system's performance. In this way, there is no need to collect and analyze large amount of information about how users interact with the system to later incorporate (manually) in the program. The system performs this task itself.

1993 Plans

The CID system itself will be fully evaluated in the context of the Space Station Freedom Program Requirement Document. This evaluation will consist of cognitive science experiments to be carried out with the Space Station Level I personnel at NASA HQ. Those experiments will be designed to: (1) analyze the way people use an advanced interaction media such as CID to browse and retrieve technical information, and (2) test and validate the CID prototype already developed.

Research will continue on studying several techniques to enhance CID capabilities. To help navigate in large documentation systems, a graphical table of contents browser has been added, which displays the hierarchical organization of the whole document. Observations of informal use of this browser have demonstrated its success in aiding the user to find desired information in the CID. We plan to extend this capability by developing a general graphical browser that displays any type of relation between portions of a document, not only hierarchical relations. A semantic correlation measure between documents is under development and will be integrated within the above graphical browser. This semantic correlation measure will be helpful in comparing and revising large documentation systems. We are also studying a concept clustering approach to organize context attributes into context classes. In the future, we will extend context-sensitive information indexing and retrieval capabilities for any kind of document (text, graphic, sound, video).

Finally, we will study the integration of CID within ongoing research at ARC, exploring issues in knowledge capture, diagnostic procedure retrieval, and other concepts related to on-line intelligent assistant systems.

Milestones

1992:   - Demonstrate and evaluate the CID system with SSF Level I personnel at NASA HQ.
         - Study context clustering and semantic correlation techniques.
         - Develop and evaluate semantic graphical navigation tools
         - Utilize CID as a research facility for other projects at Ames.
1993:   - Demonstrate CID on another application such as the on-board documentation of the Space Shuttle.
         - Study and integrate full multimedia user-interface into CID.
1994-95: - CID fully integrated and operational for NASA documentation management and maintenance.

Intelligent, Communicating Agents Project

Nils J. Nilsson
Robotics Laboratory, Department of Computer Science,Stanford University

"Intelligent communicating agents" are artificial intelligence (AI) programs that exhibit autonomous behavior in complex environments behavior that requires knowledge-based perception of environmental signals and/or data, and goal-directed reasoning to compute and execute appropriate actions in real time. Our work on these topics during the past year has been concentrated in two broad areas, namely 1) PhD dissertation work on fundamental techniques required for efficient planning, reasoning, language use, and agent interaction; and 2) work on a "teleo-reactive" language for controlling the actions of autonomous agents. The following PhD dissertations were completed:

Automatic Abstraction In Planning [Christensen, 1991a]

Traditionally, abstraction in planning has been accomplished by either state abstraction or operator abstraction, neither of which has been fully automatic. We present a new method, predicate relaxation, for automatically performing state abstraction. Predicate relaxation generates abstraction hierarchies that, for some domains, can be more useful than those generated by previous abstraction mechanisms.PABLO, a nonlinear hierarchical planner, implements predicate relaxation. Theoretical, as well as empirical results are presented which demonstrate the potential advantages of using predicate relaxation in planning. Relaxed predicates can also be used by PABLO to achieve a limited form of reactivity, whereby an executable sequence of actions is constructed in case of interruption.

We also present a new definition of hierarchical operators that allows us to guarantee a limited form of completeness. This new definition is shown to be, in some ways, more flexible than previous definitions of hierarchical operators. The ability to plan using such operators has been incorporated into PABLO.

Finally, a Classical Truth Criterion is presented that is proven to be sound and complete for a planning formalism that is general enough to include most classical planning formalisms that are based on the STRIPS assumption.

Some of results related to this dissertation were also reported in [Christensen, 1991b]

A Justification-Based Theory of Explanation [Paek, 1991]

A notion of explanation has been a fundamental concept for various artificial intelligence problems. Despite its fundamental importance, little has been done to provide a formal theory of explanation. Here we present a formal and unifying theory of explanation which is justification-based. Given a background theory and an observation, we will define when a proposition explains the observation with respect to the background theory. Our approach states that the addition of an explanation to the background theory should provide a better justification for the observation in question than the background theory alone. Our approach also provides various ways to define preference criteria on explanations. As we apply our theory of explanation to a particular domain, domain-specific characteristics allow us to further refine and develop the theory; we will explore the role of this domain-specific information in the areas of causal reasoning and plan recognition. In both areas, we will focus on the phenomenon called causal asymmetry which results because two causes of an observation interact differently than two consequences of a common cause. The justification-based approach provides a better qualitative account of causal asymmetry than previously proposed formalisms and also demonstrates its generality by showing that it produces the desired results in plan recognition.

Universal Attachment: An Integration Method for Logic Hybrids [Myers, 1991]

In this dissertation, we present the integration method of universal attachment for use in constructing logic hybrids. Universal attachment is based on links, referred to themselves as universal attachments, that connect expressions in a logic to programs and data in arbitrary representation languages. Deduction serves as the primary reasoning mechanism in universal attachment systems. However, when expressions with attachments are encountered during the inference process, the corresponding attached data and programs are used to directly evaluate those expressions. Universal attachment has several advantages over other attachment-based integration methods, including greater expressiveness and modularity.

Universal attachment is defined formally as an inference rule in an extended logical framework and is shown to satisfy an appropriate condition of soundness. The completeness of the rule is established for first-order theories on the condition that the set of attachments in use satisfies a simple closure constraint. In contrast, other attachment methods are shown to be inherently incomplete.

The dissertation also addresses more practical issues related to the use of logic hybrids. A set of policies is presented that provides guidance in exploiting the expanded representational and inferential capabilities of logic hybrids. These policies are based on experimental results obtained using an implemented universal attachment system. A procedure for automatically generating universal attachments is also defined. The procedure uses an extension of logic compilation techniques to produce new programs and attachments to those programs from a base set of previously defined attachments and programs. It is shown that the generation procedure can produce sizeable collections of useful attachments and as such is a valuable specification tool for users of attachment systems.

Distributed Agents

Andrew Kosoresow is currently working on analyzing the relative efficiency of a distributed system of agents as compared to a system with centralized control. Having investigated several other domains that would be suitable for such scheduling systems, including agent task planning, job shop scheduling, and demographic simulation, he is currently concentrating on the domain of action scheduling for messenger robots. In this domain, a set of robots operate on a graph servicing delivery requests that are generated sequentially over time. Depending on whether the robots' desired performance metric is local or global, their goal is to minimize either the personal or global resource expenditures, be it distance traveled or time.

Several factors affect the robots' performance. These include communication bandwidth and topology, the amount of information at each robot's disposal, and the compatibility of the robots' goals. When such systems of robots have been defined, it is then interesting to compare the relative performance of the different systems, as well as to compare their performance to that of a centralized control system. While benchmarking provides experimental results for particular models, it would be useful to compile a set of theoretical metrics which would give the designer bounds on the impact of various design choices on the performance of the system. Systems of cooperating robots that seek to achieve mutual or non-conflicting goals are of particular interest: is the additional overhead of communication and cooperation worth it given the advantages of a distributed system?

In the last year, Kosoresow has developed the model for the messenger robot domain. In this domain, the robots' goals are to minimize the distances they travel and the response time for the requests. (Generally, it is not possible to optimize both and a trade-off is required.) He has related this problem to the Vehicle Routing Problem and the K-server problem and has worked on extending instances of these problems to cover this particular problem instance. Currently, Kosoresow is concluding his work on bounds for the centralized model and is beginning to work on the case where the robots have distributed control with limited information about the world. The robots only have limited knowledge of the forthcoming requests and of the locations of the

other robots. The most immediate goal is to determine how well the robots would work as compared to having a centralized planner in this restricted domain.

Future work consists of extending the theory to cover further variants of the distributed model by considering the effects of varying system parameters such as the communication bandwidth, the communication network topology, and the compatibility of robot goals. The goal is to form a computational theory of distributed agent complexity. Kosoresow plans to subsequently test the results empirically in order to see how the results hold up in practice and then to check the theory on another domain to see how well it applies.

Teleo-Reactive Language Task

In order to act appropriately in a dynamic and uncertain world, agents will need programs with a high degree of run-time conditionality. We have been working on formalisms in which to express such programs for the past three years or so. We take the view that it will prove impractical for an autonomous agent either to learn a complete program of actions or to generate a complete program through automatic planning methods. So, we conclude that such programs will in large part need to be written by human programmers. Nevertheless, we think that such programs can be incrementally modified by learning and planning methods, and furthermore, that such incremental modification methods will be necessary components of useful and robust autonomous agents.

Thus, three rather separate factors influence the design of a formalism in which to express programs for converting perceptual information and internal knowledge into action, namely ease of programmability, ease of modification by learning methods, and ease of automatic plan generation. Our first attempt at inventing such a formalism resulted in what we called action networks [Nilsson, 1990]. During this reporting period, we developed another, more powerful formalism based on what we call teleo-reactive (T-R) trees. A paper describing this formalism was written and has been submitted for publication [Nilsson, 1992]. The following abstract of this paper provides a summary of the main points:

New ideas are presented for computing and organizing actions for autonomous agents in dynamic environments in which the agent's current situation cannot always be accurately discerned and in which the effects of actions cannot always be reliably predicted. The notion of "circuit semantics" for programs based on "teleo-reactive trees" is introduced. Program execution builds a combinational circuit which receives sensory inputs and controls actions. These formalisms embody a high degree of inherent conditionality and thus yield programs that are suitably reactive to their environments. At the same time, the actions computed by the programs are guided by the overall goals of the agent. The paper also speculates about how programs using these ideas could be automatically generated by artificial intelligence planning systems and adapted by learning methods.

We have experimented with several programs based on the T-R tree formalism. These experiments have been performed using the "botworld" simulator designed and implemented by Jonas Karlsson [Karlsson, 1990]. (We have since written a new version of botworld that runs on several different workstation models [Teo, 1991].) Experiments to date indicate that the T-R formalism is a convenient and effective one for writing agent action programs. Experimental comparisons of this formalism with others have not yet been made, although several similarities and differences are described in [Nilsson, 1992].

We also explored several different "delayed-reinforcement" learning methods during the year. A description of the work performed has been written in draft form [Nilsson, 1991a; 1991b]. In summary, we proposed clustering methods for input generalization and methods for learning T-R trees. Experimental investigation of these methods is now in progress using three different domains, namely the botworld simulator, a range-sensing mobile robot simulator, and (eventually) an actual physical robot. We are optimistic about our ability to develop practical learning and planning methods for the incremental modification of T-R programs.

Constraint-Directed Scheduling and Planning of Space Missions
Stephen Smith, Nichola Muscettola, Carnegie Mellon University

The goal of this project is to develop representations and problem solving techniques for the effective solution of space mission management and operation problems. Space mission operation requires effective allocation of resources to competing mission goals (scheduling) combined with the satisfaction of complex physical constraints and the expansion of system reconfiguration activities (planning). Our approach is embodied in HSTS, an integrated planning and scheduling architecture. HSTS provides a modular domain description language for representing physical constraints at different levels of abstraction, a temporal data base for flexibly representing candidate goal activities and networks of system reconfiguration activities over time, and a problem-solving methodology based on constraint posting that enables opportunistic solution development. HSTS has been applied to the generation of short-term, executable schedules for the Hubble Space Telescope (HST).

1992 Progress

Our work during 1992 is concentrated in three areas:

1. application of the HSTS framework to an additional space mission scheduling domain;
2. development of heuristic scheduling strategies within the HSTS architecture that exploit knowledge of dominant problem constraints to improve the quality of generated solutions;
3. investigation of tradeoffs (both in terms of computational performance and solution quality) relating to the degree of coupling in problem solving at different levels of abstraction.

Our progress to date and plans for the remainder of 1992 are summarized below. We are exploring the application of HSTS to a broad class of space mission domains. We continue to develop our research in the initial HST domain, by interacting closely with the Space Telescope Science Institute (STScI) to acquire domain knowledge and scheduling problems and to receive feedback on our results. At the same time, we have started to tackle a second domain: the Submillimeter Wave Astronomy Satellite (SWAS) observatory mission, currently set for launch in 1995. This domain provides two opportunities: (1) a means of demonstrating the generality of our approach across several mission planning domains and (2) a basis for more rapid transition of our results into operational use. An initial analysis of the SWAS domain, conducted in collaboration with a principal astronomer associated with the mission, has identified some necessary extensions to the types of constraints handled by HSTS, e.g., the inclusion of global constraints on power. In 1992 we plan to demonstrate a solution to the SWAS problem, developed on an initial model of the domain.

With respect to constraint-directed scheduling strategies, we are pursuing two types of approaches. On one hand, we are investigating multi-perspective scheduling strategies aimed at balancing a critical short-term scheduling tradeoff of minimizing setup time while minimizing rejection of temporally constrained candidate goals. Under this approach, schedules are incrementally built through selective use of a set of complementary activity selection and placement heuristics; determination of which heuristics to use at any point is based on the perceived difficulty of attending to each conflicting objective in the evolving solution state. In the HST domain, we have produced preliminary experimental results that indicate the utility of this approach. It was shown to perform better than a fixed approach quite similar to that employed in SPSS, the operational scheduler currently used at STScI for short term scheduling. We intend to broaden the approach to incorporate additional short term scheduling objectives, and explore integration with the contention-based focusing techniques described below.

We have also developed a new, constraint posting approach to scheduling, referred to as conflict partition scheduling (CPS). The approach is driven by recognition of resource capacity

bottlenecks - time periods where there is contention among goal activities for the same resource capacity. CPS proceeds by iteratively identifying resource capacity bottlenecks (accomplished through use of a stochastic simulation technique) and acting to lessen the level of contention by posting ordering constraints among the activities competing for capacity at the most severe bottleneck. The iterative process continues until no capacity conflicts remain, at which point a final schedule has been determined. Our goal is to experimentally demonstrate the viability of CPS for solution of large and highly constrained space-based observation scheduling problems. Experimentation performed to date has compared CPS to two state-of-the-art scheduling algorithms - micro-opportunistic search and min-conflict iterative repair - on a readily available standard set of manufacturing job-shop scheduling problems. The results obtained show a clear superiority of our approach. MIN-CONFLICT, this approach is currently used in the SPIKE long-term scheduling system for the HST. The next steps will involve the determination of scheduling problem benchmarks for the space-based observation scheduling domain and the continuation of the comparative experimental analysis on such benchmarks. We expect to do this in both the SWAS and HST domains.

Finally, we are investigating the impact of various degrees of coupling in the problem solving that takes place in HSTS at different levels of abstraction. Problem solving in the HST domain, for example, currently utilizes an abstract model for selecting and sequencing goal activities and a detailed model for constructing the detailed network of setup activities necessary to ensure executability of goal activities. We plan to evaluate the leverage gained or lost by these abstractions under both a strict, top-down interaction scheme and a tightly integrated iterative coupling.

1993 Plans

By the end of 1992, the investigation conducted on the SWAS and HST observation scheduling domains will have clarified the most important aspects of both the domain structure and characteristics of typical classes of scheduling constraints that arise in the space-based observation scheduling domain. In 1993, we will concentrate our efforts on demonstration of direct use of planning and scheduling systems using the HSTS technology in the operation of actual missions. In the SWAS domain, we will provide a validated and detailed model of the spacecraft and implement a scheduling tool directly usable by the SWAS Science Team both for mission planning and for mission operation. In the HST domain, we will revise and update our initial model of the HST operating environment through direct interaction with STScI. On this new model, we will demonstrate the operationality of the scheduling techniques developed so far. We expect the refinement of the HSTS technology and of the scheduling and planning techniques to continue, as we increase the complexity of the domain models and we broaden the classes of constraints taken into account.

1994 Plans

In 1994 we will continue our effort to transfer the HSTS technology into actual operation. We will extend the scheduler for the SWAS mission to include reactive schedule revision capabilities, to respond to unexpected events. We will also provide a prototype for an operational tool directly usable at STScI; we will demonstrate how the tool will further enhance the capabilities provided by the current operational software and provide valid help for increasing mission effectiveness.

Milestones

1992   - Initial HSTS solution in SWAS domain
       - Experimentation with scheduling strategies and inter-level coupling complete
1993   - HSTS deployed for SWAS scheduling
       - scale-up to full HST model complete
1994   - HSTS deployed for support of operations at STScI

Acting to Gain Information
Stanley J. Rosenchein, Teleos Research

Teleos Research is carrying out a three-year program of research aimed at developing new techniques for constructing real-time systems that act to gain information. This is part of a larger research program focused on the design and implementation of intelligent reactive agents, which has had ongoing support from NASA and from agencies within the Department of Defense. Our work in this larger program has included the development of techniques for planning and acting in intelligent agents. This work has brought out the importance of perceptual action and inspired the new research direction of studying acting to gain information. This project seeks to extend previously developed techniques and to apply them to the problem of actively acquiring information from the environment in support of achieving particular goals.

Our work consists of several interrelated strands of research. First, there is work on the theoretical foundations of information-gathering strategies. This work aims to model the information states of agents and the means by which these information states are altered in goal-directed ways by actions of the agents themselves. This research has also been concerned with developing extensions of programming methodologies for specifying reactive goal-seeking systems to handle the specialized problems of information gathering.

1992 Progress

This year our theoretical investigations have also included modeling information gathering among groups of agents. This work, carried out by project member Stan Rosenschein, has begun to relate logics of mutual belief and joint action, such as those developed by Cohen and Levesque, to situated-automata work on the compilation of multi-agent perception and action systems. In a related area, project member David Chapman has been extending the theory of situated automata to account for coordinated multi-agent action that is subject to interactional rules. The thrust of the work is to develop a concept of rules that can be violated when an agent is unable or unwilling to comply with them and protocols for agents to communicate with each other concerning the consequences of such violations. A paper based on Chapman's work has been submitted to the AAAI Conference.

Several aspects of our research have been motivated by physical tasks that require information gathering in real time. Intelligent strategies for such tasks (e.g., visual tracking of moving objects and searching for hidden objects by looking inside containers and by digging in sand) are relevant to a variety of autonomous space systems and offer many challenging test problems. In this context, we have studied selected topics in active vision, such as the control of gaze and point of view, from the perspective of reactive information gathering. This work makes use of the experimental agent-architecture testbed being developed under a related project. This testbed includes actively controllable sensors mounted on a gantry and provides the ability to carry out controlled experiments on reactive information gathering.

In addition to studying the acquisition of short-term information from the environment via controlled perception, we have also been investigating the acquisition and use of long-term, semi-permanent information via learning, especially learning how to act -- including learning information-gathering actions.

In earlier work, we studied the use of reinforcement-learning techniques, both in the context of a simulated world and in a real robotic system. During this past year, Leslie Kaelbling extended her empirical investigations of learning techniques for embedded systems by refining the necessity and sufficiency operators used to guide search in a generate-and-test reinforcement-learning algorithm. The refined operators were shown to perform well in extensive empirical trials. In addition, Kaelbling has studied properties of generalization over large output spaces in reinforcement learning. Empirical tests show that the TEAM strategy described in the literature works very poorly because of an inability to explore the space of possible outputs in an organized way. The CASCADE method developed by Kaelbling, on the other hand, solves this problem,

leading to good empirical learning performance and a reduction in computational complexity from the original solution.

1993 Plans

During 1993, we plan to bring together the multiple strands of our research and demonstrate sophisticated techniques for guiding information-gathering actions in real time. We plan to continue our theoretical work, including work on groups of interacting information-gathering agents, and to have completed at least one significant demonstration system, which we hope to document in a videotape. We expect this demonstration to combine techniques of real-time planning, reactive control, and autonomous learning and to demonstrate relevance of our approach to task requirements in complex domains, including real-time perception for space robotics, and non-robotic applications, e.g., on-line diagnosis and equipment maintenance. We will also be documenting the results of our research for publication in journals and for presentation at scientific meetings.

Milestones

1992: - Analyze patterns of reasoning used in implementing information-gathering strategies and define new abstractions, programming constructs and idioms, that exploit these patterns to simplify implementation.
- Implement information-gathering strategies using the new abstractions and refine programming tools based on experience of applying them to the target tasks.
- Investigate the use of learning techniques to allow an agent to learn perceptual action strategies from its experience in the world.
1993: - Apply programming tools to more complex task set, including tasks from robotic and non-robotic domains (e.g., diagnostic tasks).
- Test learning techniques on target tasks. Compare learned strategies to previously hand-coded ones.
- Integrate learning techniques with programming tools to allow the development of systems that are partially programmed.

MACHINE LEARNING

Icarus: An Integrated Architecture for Intelligent Agents
Pat Langley, John Allen, Wayne Iba, Kevin Thompson

The Icarus project is developing a framework for the control of autonomous intelligent agents. The goal is a system that covers a broad range of cognitive behavior, including the recognition of physical objects and events, the generation of plans, and the execution of motor skills. The long-term aim is to integrate these aspects of cognition into an architecture that can interact with an unpredictable environment and acquire knowledge from its experience. Potential applications include NASA missions in which an autonomous agent functions in lieu of an astronaut, as in planetary exploration or the execution of deep-space scientific experiments. Such an agent would be suited to any environment for which it is costly and/or hazardous to send humans (e.g., machine diagnosis/repair, substance cleanup in radioactive areas, or "trouble-shooting" in a planetary habitat).

1992 Progress

Our preliminary designs for Icarus assumed that long-term memory was organized as a hierarchy of probabilistic concepts, which it used to store knowledge of physical objects, plans, and motor schemas. The agent would then use a process of heuristic classification to retrieve relevant knowledge in new situations, to insert experiences into memory, and to modify the structure of this memory. The basis for the entire architecture was Cobweb, an algorithm for probabilistic classification and learning developed by Doug Fisher, an Ames grantee at Vanderbilt University.

However, we encountered recurring difficulties in using Cobweb's mechanisms for the more complex knowledge structures needed in Icarus. We found that, although the acquired hierarchies had good predictive ability, they were often poorly structured in that similar experiences were structurally distant from each other in memory. In response, we developed an alternative algorithm, called Arachne, that we felt would be less sensitive to the order of training instances and would produce hierarchies with more coherent structure. Comparative experiments revealed some improvement over the Cobweb approach, but not as much as we had hoped. We also carried out separate experiments with Cobweb_R, a related algorithm that handles relational descriptions; the results here were also disappointing, suggesting that the use of probabilities in matching provided no advantage over a simpler scheme.

Our designs for Icarus included three major components: Labyrinth, which classifies structured descriptions (including objects, events, and plans) and stores abstractions of them in long-term memory; Daedalus, which generates plans and improves this ability with practice; and Maeander, which acquires motor schemas that it can use to recognize or execute actions. Each of these systems uses a variant of Cobweb as its central routine for classification and concept formation. Despite the problems with Cobweb mentioned above, during this year we continued to extend and study each of these systems. Our work included studies of the effect of background knowledge in Labyrinth , evaluation of Daedalus and Maeander as psychological models, and systematic experiments with the Maeander system.

In addition to studying the components of Icarus, we also made some progress toward integrating them into a single architecture. This led to one workshop paper and to various internal design documents. To sidestep our problems with Cobweb, we designed a reduced architecture that did not include probabilistic representations, a hierarchical memory, or learning abilities.

We have made progress in revising our approach to planning to include storage and retrieval of complete plans. This extension of Daedalus, called Talus, can operate in three separate modes. Like its predecessor, Talus can use plan knowledge as separate `control rules' for selecting actions; this provides flexibility but requires expensive retrieval at each decision point. Alternatively, the system can use stored plans as opaque `macro-operators' that reduce retrieval

costs at the expense of flexibility. Finally, Talus can take the middle path of adapting plans analogically, which gives moderate retrieval costs and flexibility but requires checking plan components for relevance. In addition, the implementation provides support for planning in domains with numeric descriptions, including simple manipulation and navigation tasks. Our work on Talus has led to one workshop paper describing the system and preliminary experimental results.

Our frustration with complex systems that we could not understand also led us to spend time away from the main thrust of the project on the theoretical analysis of related but simpler algorithms. In particular, we carried out one average-case analysis of an algorithm for inducing simple decision trees and another of an algorithm for learning simple probabilistic descriptions. Both studies identified factors affecting learning rate, made predictions about behavior in synthetic domains, and confirmed these predictions with experiments. Experience with search control in Daedalus had also generated some ideas about flaws in depth-first search, and another average-case analysis identified the conditions under which a random, iterative search algorithm outperforms the more traditional systematic approach .

1993 Plans

Our experiences over the past year have modified our plans for Icarus. For the near term, we envision a system that stores knowledge of objects, plans, and motor skills in a simple list, so as to avoid issues in memory organization. Retrieval will involve a simple all-or-none matching process, and learning will be reduced to the collection of statistics to control plan reuse, monitoring, and execution. Our priorities for the architecture include:

1. Extending Talus to collect statistics about the frequency of plan reuse, and to use this information in plan generation. In particular, on retrieving a plan whose components tend to be reused successfully, the system should treat the structure as a macro-operator. In contrast, on accessing a plan whose components are seldom reused, it should use the structure only for local search control. In cases where statistics are ambiguous or insufficient data exist, Talus should test the plan components for relevance and use them if appropriate.

2. Implementing an initial version of Icarus that interleaves planning, execution, and perception. The central issue here involves when to halt deliberation in order to execute and/or sense. The system would collect and store conditional probabilities on plan components for use in these decisions. In cases where backtracking is unlikely, Talus could start to execute a plan before its complete specification. In situations where violated expectations are unlikely, it could bypass perception entirely.

3. Designing mechanisms to handle interruptions and changes in goal priorities. In this scheme, classification of a new state could cause the retrieval of a stored problem that differs from the one currently being pursued. If the retrieved problem has higher priority than the currently active one, the latter would be suspended and the agent would pursue the more urgent one. Once this has been handled, control would pass back to the original problem, unless another one has taken over in the meantime.

We continue to explore NASA domains in which to test our ideas on recognition, planning, control, and learning. The best contender involves planning and execution in the Reaction Control System, which is responsible for attitude changes in the space shuttle. We have completed a preliminary analysis of this task, and we hope to gain some experience with Talus on realistic problems sometime next year.

Milestones

1992: - Modify the Talus planner to collect statistics on plan reuse and to use this information in selecting planning modes.
- Begin to interleave processes of planning, execution, and perception.
- Design mechanisms for handling interruptions and goal conflicts.

1993: - Complete initial integrated version of Icarus that interleaves planning, execution, and
        perception, and that can handle interruptions.
       - Begin encoding of knowledge for a NASA testbed, such as planning
        and execution for the Reaction Control System.
1994: - Begin evaluation of Icarus on realistic problems involving the
        planning and execution of the Reaction Control System; refine
        the architecture in response to these studies.
1995: - Complete evaluation of Icarus on problems involving the Reaction
       - Control System; improve robustness of the architecture.
1996: - Incorporate Icarus software into a prototype package for use
        with the Reaction Control System by Mission Control.

Heuristic Optimization for Planning and Scheduling
Steven Minton, Monte Zweben, Andrew Philips, Eugene Davis, Michael Hall

The primary goal of this work is to improve upon current AI-based methods for planning and scheduling. One of the limitations of current AI systems for planning and scheduling is that considerable domain-specific expertise must be programmed into them. Generally, this expertise takes the form of heuristics. Deriving appropriate heuristics is a time-consuming process that requires specially-trained personnel. To address this problem, we are focusing on two related approaches. First, we are analyzing and comparing current planning and scheduling techniques, with the aim of identifying generic (i.e., domain-independent) heuristic methods that are appropriate for planning and scheduling problems. In most cases, this requires analyzing previously-developed heuristics to identify the types of problems where these heuristics are most useful. Second, we are developing machine learning techniques that can automatically select appropriate heuristics for a task based on experience.

1992 Progress

In the past year, we have analyzed several generic heuristics that have proved promising. The MIN-CONFLICTS heuristic, which we developed in an earlier phase of this project, has been employed in a variety of scheduling applications. The heuristic guides the scheduling process by minimizing the total number of constraint violations in a schedule. This work has been used by researchers at the Space Telescope Institute for improving the Hubble Space Telescope Scheduler. Our collaboration is continuing as the project is extended to other scheduling applications. We have recently expanded our previous understanding of this heuristic by completing additional theoretical work and carrying out several empirical tests. By improving our understanding of the heuristic, we can make better predictions as to when the heuristic is appropriate.

In a related effort, Minton and Zweben began investigating the tradeoff between the informedness of repair-based scheduling heuristics versus their computational complexity. We are trying to dissect repair -based heuristics into their most elemental parts such as breadth, depth, and evaluation criteria. Breadth addresses the number of choices that a particular repair considers and depth is the degree of lookahead used to select between these choices. The evaluation criteria is the knowledge used to select between possible schedule repairs. Not surprisingly, our experiments suggest that no one heuristic applies to all problems. We found that a heuristic that uses no lookahead performs better than a more informed and computationally complex lookahead heuristic on a specific Space Shuttle scheduling problem. However, lookahead performs better on other problems. In 1992, we hope to further investigate this tradeoff and to develop techniques that allow systems to predict the utility of more informed heuristics.

In the past year, Zweben and Davis also developed a version of Plausible Explanation-Based Learning (PEBL) that allowed the Space Shuttle Ground Processing Scheduling System to learn when a lookahead heuristic would be more useful than one that does no lookahead. This learning system reviews the effectiveness of the no-lookahead repair and when it appears to lead

the search awry it dynamically adopts the lookahead technique. Experiments indicate 20%-40% improvement in search performance.

Minton et. al. also completed both a theoretical analysis and an empirical analysis of partial-order planning, a heuristic technique that is employed in many AI planning systems. This work revealed several factors that are required for this heuristic to be effective. Moreover, in the process of carrying out the analysis, we discovered that a new search technique may be especially appropriate for planning problems. (This search technique is also being explored by Pat Langley). Preliminary results were published at the International AI conference in the summer of 1991, and we expect to complete a journal-length paper in the spring of 1992. (This project was the result of joint work by Minton, Drummond and Bresina).

1993 Plans

In developing machine learning techniques to automatically refine generic heuristics for particular tasks, our initial results have also been encouraging. In previous work, Minton et. al. developed the PRODIGY/EBL system for automatically learning heuristics from experience. This system has been distributed to many researchers around the world, and has attracted considerable attention. In work published in 1991, Minton and several colleagues (now at Carnegie-Mellon University, the University of Washington, and the USC Information Sciences Institute) have continued to identify the strong and weak points of the PRODIGY/EBL system, and to suggest further improvements to the original system. We now believe that it is time to move past this work and develop a new system, based on what was learned with PRODIGY/EBL. Minton is, therefore, working on a new system, ASAP-CSP, which is designed to refine generic heuristics for scheduling and constraint satisfaction problems. ASAP-CSP builds on the PRODIGY approach, but is a more practical system in several respects. The primary advance is that the system is designed for more specialized problems, and thus a more specialized optimization theory can be employed. This is expected to dramatically increase the number of useful heuristics that the system can produce. Work is expected to continue on this system throughout 1992, with the goal of conducting empirical tests in early 1993.

PEBL research will be extended in 1993 in two major ways. First we will integrate an abstraction mechanism that will learn statistically useful abstratctions. Constraints will be organized into abstraction levels that will allow the scheduler to ignore certain constraints until others at a higher level of abstraction have been repaired. Unlike current work in abstraction, we endeavor to learn statistically useful abstractions across a suite of problems rather than a set of abstraction levels that are guaranteed to be montonically solvable. Monotonically solvable abstractions guarantee that once higher level abstraction levels are solved, they will never have to be addressed repeatedly. We believe that the conditions under which montonically solvable abstraction levels can be found are unlikely to be present in realistic scheduling problems because of the inter-connectedness of constraints.

The second topic of PEBL research involves learning more robust schedules. In a dynamic operations environment, schedules are often modified because of priority expectations about the duration of activities, the availability of resources, or the introduction of unexpected tasks are inaccurate. This phenomenon is worsened by scheduling systems that attempt to optimize, because a near-optimal schedule is not necessarily one that is flexible enough to easily adapt to exogenous events. We plan to develop learning techniques that monitor schedule execution and isolate poor assumptions. These assumptions will then be modified, hopefully, yielding more robust schedules that will require less modification during schedule execution.

Milestones

1992:   - Machine Learning Conference paper submitted on the application of PEBL to iterative repair.
          - AI Planning Conference paper submitted investigating the tradeoff between repair informedness and computational complexity.
          - Invited AI Journal paper on MIN-CONFLICTS.

- Invited AI Journal paper on PEBL.
1993: - First empirical results of the performance of CSP-ASAP.
       - First empirical results of PEBL-Abstraction
       - Development of techniques that predict how informed repair-heuristics should be on a suite of problems.

Adaptive Knowledge Bases
Smadar Kedar, Kathleen McKusick
Lonnie Chrisman, Carnegie-Mellon University

The long-term goal of this project is to enable complex integrated systems to improve with experience. Many of NASA's tasks require complex planning, scheduling and control. Often these tasks need to be integrated and coordinated. To build such complex systems, multiple knowledge bases and procedures are encoded to perform multiple functions. One critical issue is how to detect and correct errors and inconsistency in the knowledge bases that could lead to system failures. Currently if there are system failures, system builders refine and debug the knowledge bases manually in an ad-hoc and error-prone approach. The refinement process is also time-consuming, often requiring much more time than the initial encoding of the knowledge. The Adaptive Knowledge Bases project is developing tools to assist users to systematically and more automatically debug and refine multiple knowledge bases.

Our approach is to use as our testbed an existing state-of-the-art integrated system for planning, scheduling, and control. We use the Entropy Reduction Engine (ERE), described elsewhere in this report. Our methodology is to develop various learning algorithms to address problems and inconsistency discovered in the multiple knowledge bases of ERE, and implement and test them within ERE. This leads to general principles applicable to other systems and domains.

1992 Progress

Mutual Theory Refinement: In building the ERE integrated system, knowledge of the possible actions of an agent in the world have been hand-coded into operator descriptions. Related knowledge has also been hand-coded as domain axioms describing possible and impossible situations in the world. Both operators and domain axioms are used in planning and execution (control). Through our experience in integrating both components we have discovered that when planning uses incorrect operator descriptions or domain axioms, invalid plans may result. And since the expected outcomes of such plans may differ from the actual outcomes in execution, the agent may be unable to achieve its intended goal, leading to system failures.

In 1992 we completed version II of MTR, the prototype learning system for refining knowledge bases. Version II augments the previous version with additional refinement methods: adding missing domain axioms using the operator descriptions as a guide; correcting incorrect domain axioms; adding missing operator effects; removing extra operator effects. One novel research contribution of this system is that it uses multiple incorrect and incomplete knowledge bases to refine each other. It can detect when a knowledge base is insufficient for refinement, and degrade gracefully, using less powerful machine learning methods such as induction or rote learning. Most previous knowledge-intensive machine learning methods required a complete and correct knowledge base in order to refine error full knowledge bases. Yet this is too strong an assumption for real-world domains, since knowledge bases will always be incomplete and / or incorrect when modelling the real world.

Utility of Refinement: Another research result in 1992 is a demonstration of how knowledge base refinement improves performance. We completed preliminary ˌperiments which demonstrate that knowledge base refinement improves system accuracy -- ˌ ˌ's ability to achieve its goals. However, the refinement degrades, rather than improves, the efficiency of the system, due to increased match and search cost of planning and execution in ERE.

Interactive Knowledge Acquisition: In 1992, we added an intelligent user interface to MTR, enabling interactive knowledge acquisition. The interface suggests possible refinements, and allows the user to choose among them. The system can also run in a fully automated mode. We implemented a flexible control structure which considers alternative causes of an error, and alternative refinements. The system uses heuristics which enable it to choose among various refinements.

Applications: We completed three detailed investigations into potential applications for our machine learning algorithms in 1992. We are currently exploring the APT telescope application being started by Drummond et. al. as an application of our learning research.

1993 Plans

Mutual Theory Refinement: We plan to add additional refinement methods to MTR. We plan to concentrate on inductive methods. One proposed inductive refinement method would refine the operators, adding new non-deterministic outcomes when such outcomes are observed but not anticipated. In the course of time if such outcomes are correlated with certain preconditions, the method would induce a causal relationship between preconditions and outcomes, and attempt to confirm the causality by experimentation.

Utility of Refinement: We plan to extend our experimentation on the utility of knowledge base refinement. A proposed method would monitor increased accuracy and decreased efficiency, and, without affecting accuracy, will attempt to improve efficiency by approximating the knowledge bases in an informed manner.

Interactive Knowledge Acquisition: We plan to continue investigating the area of interactive knowledge acquisition. We plan to port the user interface to a system for refining procedures in the Procedure Management and Maintenance Project, described elsewhere in this publication. The synergy between machine learning and knowledge acquisition is new, and we expect to put more of our emphasis on this area in the coming year.

Applications: We plan to focus on one application, in order to demonstrate the feasibility of using adaptive knowledge bases to reduce errors and increase the productivity of complex NASA operations.

Milestones:

1993: - Design and implement additional refinement algorithms in the MTR system.
- Test the system in a simulated environment (NASA TileWorld).
- Design, implement, and test a prototype of the system in the chosen NASA application.
- Expand the intelligent user interface, and port it to a procedure maintenance system.
1994 - Complete full integration of MTR into the ERE system
- Develop and implement knowledge acquisition algorithms for the intelligent user interface
- Test and expand the prototype system in the NASA domain
1995 - Integrate interface and knowledge acquisition with application system
- Complete full prototype of application system
- Explore and apply system to second application area working directly with technology transfer personnel

Automatic Classification
Peter Cheeseman, John Stutz, Robin Hanson, Will Taylor

The AutoClass project is directed toward automatic discovery of probabilistic classes in large databases.

Background: In previous years, the Bayes group developed the basic theory for automatic classification of data. We have implemented this theory in the AutoClass series of programs, and are now experimenting with AutoClass IV. AutoClass takes a database of cases described by a combination of real and discrete valued attributes, and automatically finds the natural classes in that data. It does not need to be told how many classes are present or what they look like. Instead, it extracts this information from the data. The classes are described probabilistically, so that a case can have partial membership in the different classes, and the class definitions can overlap.

Recent Developments: AutoClass IV has greatly relaxed restrictive assumptions built into earlier versions. It now allows the variables within a class to be correlated, and allows the classes themselves to form a hierarchy. These extensions mean that the user does not have to carefully pick attributes so that they are independent, or decide ahead of time what attributes are relevant. An irrelevant attribute is promoted in the class hierarchy, indicating that it has no discriminating power for the classes below it. We are experimenting with ways to efficiently search the expanded class space generated by these extensions.

The Bayes group is also developing a number of specialized models for particular applications. We are now implementing a model for handling angular data, such as occurs in describing protein structures or astronomical directions. We are developing a model for taking into account correlations between neighboring values of one and two dimensioned arrays, such as occurs in spectral data and images. This neighbor correlation model will allow us to perform a much more detailed reclassification of the IRAS spectral data performed by an earlier version of AutoClass. We also implemented a technique for dealing with very large data bases via continuous random subsampling.

The AutoClass algorithm is inherently parallel. An earlier port of the core procedures to a parallel machine achieved a speed linear in the number of processors. To take advantage of this parallelism, we are porting the new AutoClass to the Connection Machine, so that we will be able to process databases that are too large for the current serial machines.

The probability theory underlying AutoClass IV has been described in a NASA technical report and in a paper presented at the 1991 IJCAI meeting. A technical report describing a classification of LANDSAT data (i.e., a set of ground-cover types) using this new version of AutoClass is also available. An extended AutoClass III, basically AutoClass IV without class hierarchies, is being prepared for public release through COSMIC in early 1992.

Milestones:

1992: - Develop and implement neighbor correlation models, and angular data models. Port AutoClass IV to the connection machine, and reprocess IRAS spectral data. Release a version of AutoClass through COSMIC.
1993: - Incremental improvements as needed. No major extensions.

Data-Driven Surface Modeling.
Peter Cheeseman, John Stutz, Robin Hanson, Bob Kanefsky, Wray Buntine, Rich Kraft

A common problem in scientific data analysis is how to combine multiple observations of the same thing to get the best composite view (model). Combining multiple scans from the IRAS satellite to give a best view of the infrared sky is one example. Combining many LANDSAT passes over the

same area to give a best map of the ground cover is another. There are many difficulties in forming the best composite view. These include such things as multiple image registration, atmospheric distortions, instrumental effects (e.g., jitter), temporal and spatial variation, etc. But their effects on an image can be taken into account by explicit modeling.

Beginning in late 1991, the Bayes group began a new project to develop the theory and practice of combining data from multiple images. The essence of our approach is not to try to mash the data together by some ad hoc algorithm. Instead we model the processes that generated the multiple views. For example, with the IRAS multiple scans, we will model the distribution of the infra-red sources in space (point sources and extended sources) that best fits the multiple scans, while avoiding the addition of detail that is not justified by the data. This process incorporates weak knowledge about the possible sources as well as detailed information about the blurring function and pointing error of the detector.

A more complex process model is provided by the LANDSAT data example, which is our main long-term goal. Many factors contribute to the observed intensity of a particular LANDSAT pixel, including the type of ground cover on the corresponding ground patch, the patch's slope, atmospheric absorption of light going to and from the patch, the sun and viewing angles, and any camera distortions. If these parameters are known for every patch on the ground, it is possible to predict the corresponding observed pixel intensity (to some noise level). The problem the Bayes group is trying to solve is the inverse problem. Given the observed intensities (pixels), we seek the most likely ground truth (parameter values) that agree with this data.

The technique under development by the Bayes group produces a model at higher resolution than the original data. As more views are incorporated into the model, the degree of super-resolution increases. This technique is closely related to computer tomography as used in X-ray interpretation, astronomy, and seismology. Normally, super-resolution causes great difficulties because of over-fitting. If too fine a model resolution is used, there will be many possible different super-resolved models that all reproduce the given data to within the noise level. We avoid this difficulty by allowing a spatial correlation between adjacent elements of the model, so that these elements are no longer independent. This spatial correlation occurs at multiple scales and is a mathematical expression of the commonly observed fractal distribution of surface properties in natural scenes.

As an initial domain, we have begun using Mars image data from the Viking orbiters. We have identified a test area where there are dozens of images of the same place, and we will build a model of this area. The model will consist of an elevation and albedo (specific reflectance) map laid out on a high resolution grid. From this model information, and information about sun and satellite angles, we can generate an image of what that area would look like from any particular view point. Inverting this process will give us the reconstructed surface model that best predicts the many images we have of that area. The reasons for choosing Mars data for our initial experiment are that the data is free, atmospheric effects are normally negligible, the surface is constant, and the camera characteristics are particularly simple. The software we develop for this test data should be useful for producing super-resolved surface models of other planetary data. The recently glimpsed asteroid Gaspra is a potential candidate. In the case of earth, it also deals with the problem of integrating information from multiple instruments.

Milestones:

1992: - Develop simple light scattering models, camera models, correlation models, etc. for the Mars data. Implement the mathematical models as a program (greatly simplified) and apply to the Mars data.
- Extension of the initial models to include the most important atmospheric effects and simple temporal models. Testing and debugging of these extensions by attempting to integrate a set of LANDSAT images of an area where ground truth is also available.

1993: - Further extensions of light scattering/surface, etc. models to remove the major sources of error, as identified by extensive testing. Evaluation of software for possible use as the major information processing tool for LANDSAT (or similar) data.

Evolutionary Tree Reconstruction
Megan Eskey, Peter Cheeseman

One of the challenging questions of interest to NASA is: "do other planets (around other stars) have life?". One way of getting some insight to this question is by studying how life originated on earth---the only planet that we know for certain supports life. One way of understanding the origin of life on earth is to work backwards from current life forms to make inferences about distant ancestors. The availability of DNA sequences, and the known high degree of conservation of these sequences, especially those that code for critical proteins, make possible the unravelling of early ancestors. For this reason we are investigating evolutionary tree reconstruction from DNA and protein sequence data.

We have developed a general theory for reconstructing paths of evolution from molecular sequence data (sequences of DNA and amino acids), and have begun the preliminary design and implementation of a system called GenesYs. GenesYs uses a Bayesian version of minimal message length (MML) encoding to reconstruct a set of highly probable evolutionary trees. The goal of our research is to improve current computer-based methodologies for phylogenic reconstruction. One advantage to our approach is that MML/Bayes provides a mechanism for the necessary trade-off between the complexity of a given model of evolutionary processes with the model's fit to the data. We intend to test our system on a range of data types, including data that has been well studied, such as cytochrome c, and hypervariable data, such as the non-coding region of human mitochondrial DNA. Ultimately, we will apply the system to RNA sequence data in the hopes that it will provide some insight into the evolutionary events leading to the accretion of complex genomes.

Milestones:

1992: - Extend the system to more accurately model evolutionary processes by adding the ability to model probabilistic ancestors.
1993: - Extend GenesYs to model the evolution of RNA viruses. Apply the system to RNA viral data.

Model-Based Learning
Wray Buntine, Richard Kraft

The goal of this research is to develop generic tools for learning from data and from partial models of the domain, and to develop the capability to rapidly develop and tailor these learning tools for particular domains.

These tools are used to discover prescriptions for predicting target variables in the domain (supervised learning, induction, or regression) or to discover models describing the domain (model learning). The input to the tools are data and prior knowledge in the form of partial models of the domain supplied by the domain expert. The partial models can be as little as recommendations like "rule sets should be good classifiers for this domain," "the underlying process is time invariant," "there is no causal link between these sets of variables," or "this intermediate variable is important for classification".

The tools are being developed for the case where the amount of data is too small for the technique of "minimizing empirical risk" to be applied. In this case, Bayesian statistics and decision theory are the appropriate theory to use in designing the learning algorithms. This theoretical guide is important because empirical, *ad hoc* development of algorithms in the past has been time consuming and is often plagued by unexplained problems. Empirical validation of the algorithms is also important to check approximations made in interpreting the Bayesian theory. We do this empirical validation by applying the algorithms to a battery of recognized learning problems taken from the literature, or manufactured problems.

1992 Progress

The IND Tree Package was prepared and released as a beta test to the research and development community. Trees are tree structured classifiers useful for classification, and tree algorithms are a well known form of supervised learning. About 30 universities and R&D laboratories in the US currently have the beta test code. The code release includes 100 pages of documentation and 15000 lines of C code and C shell scripts. Main use of the code to date has been in benchmarking, comparative studies, and comparative research on related algorithms. Extensions planned for 1992 include adding features to handle missing values, multivariate splits, and rule sets instead of just trees. The code will also be submitted to COSMIC early in 1992.

Basic research on function finding algorithms (regression for learning functions from data) was undertaken and methods developed and reported to the research community in a journal article. About 50 requests for preprints on the basic results were received from researchers in the US and Europe and several US researchers have independently begun implementation of the basic results. In addition to implementing function finding algorithms we expect to implement methods for learning Bayesian networks and to integrate these methods into the IND package.

Applications currently being investigated by the group include analysis of plume spectral data from space shuttle main engines, analysis of incidents reported to the NASA/FAA Aviation Safety Reporting System, and predicting the power usage of NASA Ames wind tunnels. Richard Kraft joined the group on 12/23/91 to work on these. Applications and implementations will continue to further progress.

1993 Plans

Research in 1993 will consist of consolidating the research on learning Bayesian networks, and function finding, and implementing basic systems for these tasks. Future extensions depend on the perceived need and the applications developed, although we might consider time series, diagnosis, spectral analysis and classification, or dynamic systems and learning control.

Milestones

1992: - Extend research on function finding (regression) algorithms for learning functions from data.
- Incorporate techniques for specifying and improving partial models into the learning system under development.
- Begin implementation of extended IND system for learning Bayesian networks and for function finding (regression).
- Continue work on NASA applications of the learning tools.
1993: - Update and maintain distributed code to improve functionality according to requests from users.
- Write journal articles reporting results as appropriate.
- Develop generic methods for rapidly generating learning algorithms from user specification of model families.
- Continue work on NASA applications of the learning tools.
1994: - Update and maintain distributed code to improve functionality according to requests from users.
- Develop methods for time-series, diagnosis, spectral analysis and classification or dynamical systems, depending on perceived need in NASA.
- Continue work on NASA applications of the learning tools.

Efficient Learning Algorithms
Phil Laird, Ron Saul

NASA software typically has a much longer life cycle than conventional software. Finding reliable techniques for writing programs that learn (i.e., adapt and improve) over these long time periods is acknowledged to be a major challenge for the machine learning field and essential for NASA's long-term objectives. Like any engineering technology, however, machine learning can progress only so far without the fundamental theory to analyze the power and limits of its methods. The purpose of the Efficient Learning Algorithms project is to apply existing theory to the development of practical algorithms and to construct new, well-understood learning algorithms that can be applied to NASA domains.

1992 Progress

This past year we designed a new algorithm for an important machine learning problem: discrete sequence prediction (DSP). The DSP problem arises in many applications where the ability to observe, learn, and eventually predict the future of a sequence of symbols is the basis for adaptive improvements. Examples include data compression, automated telemetry monitoring, adaptive game playing (applicable, for example, to pilot training), dynamic buffering, program optimization, and intelligent machine interfaces. The TDAG algorithm is a general, theoretically sound, and useful algorithm for the DSP problem. Because of our success in developing the TDAG theory and two prototypical applications using the TDAG algorithm, we postponed other research plans during 1992.

Briefly, the TDAG algorithm learns to predict by observing a stream of symbols. Other algorithms for this problem, most often based on Markov processes, have only limited usefulness because they take too long to learn. Often they are of practical use only when the number of different symbols occurring in the input is quite small. The TDAG algorithm avoids these problems by using a new representation for statistically dependent events. It is fast, uses a small amount of computer memory, and is suitable for online applications, even when the number of different input symbols is large.

During this year, we developed the theory of the TDAG algorithm and demonstrated its practicality in two applications: adaptive file compression and dynamic program optimization. Papers describing the TDAG and the experimental results were prepared and submitted to major conferences.

Adaptive file compression reduces the size of a data set by recoding it in a way that reduces redundancy. We replaced the learning element in a standard file compression algorithm (dynamic Huffman coding) with a TDAG and observed the compression of several text files. Results showed that the TDAG program gives compression ratios as high as or higher than those of licensed Unix utilities (*compress* and *compact*). This was surprising, in view of the fact the the TDAG was not designed primarily for file compression (but the other algorithms were).

Computer programs are designed to work correctly for every possible input. Some inputs, however, are much more frequent than others, and ideally a program should operate most efficiently on those problems it solves most often. In practice, however, the programmer who is designing a program does not know the distribution of the input data; even if he did, he might not know how best to reorganize the program. Dynamic program optimization reorganizes a correct program in such a way that it requires less time on average. We wrote a dynamic optimizer for Prolog (an Artificial Intelligence programming language) using the TDAG algorithm to learn patterns in program execution sequences. Using this TDAG information, the optimizer reformats the program into a functionally identical program but one in which common series of steps are combined and search is biased toward choices most likely to lead to the desired result. Our dynamic optimizer reduced program run times anywhere from a few percent on hard search problems to nearly fifty percent on highly repetitive parsing problems.

1993 Plans

During the coming year, we will explore a number of further applications of the TDAG algorithm. One of the most promising of these is a method for optimizing planning and scheduling by learning choices beneficial to the "average-case". Another application is to enable software to learn about and adapt to the human using it. Another application under discussion is dynamic optimization of CLIPS programs; if feasible, this would be a valuable enhancement to the CLIPS language, in view of the fact that NASA uses it extensively and licenses it to outside users.

Plans to prototype the CDFI algorithm, which was developed in 1990 by this project, were deferred in 1992 in order to complete the TDAG research. We plan to resume work on applying the algorithm as a low-knowledge, automatic diagnosis tool of broad applicability. With the CDFI algorithm as a basis, we will design and implement a "plug-in monitor" (PLUM) prototype. The purpose of this system is to observe any of a wide variety of controlled systems, learn the normal operating characteristics of the device (with very little explicit built-in knowledge) and then to watch the device during its operation and detect unexpected changes due to malfunctions or gradual drift. Various control devices, both stable and unstable, will be simulated in order to evaluate the practicality of the method. If successful, a software prototype will be released.

Milestones:

1992: - Preliminary report on new speedup-learning methods. (Completed successfully.)
       - Initial PLUM implementation
1993: - Release Common LISP and C-language TDAG implementations.
       - Explore further TDAG applications.
       - Write the initial implementation of PLUM and apply it to
         a device-control system.

Machine Learning and Planning in Dynamic Environments
Jaime G. Carbonell, Matthew Mason, Tom M. Mitchell, Carnegie Mellon University

Learning from Experimentation in the PRODIGY planning architecture has proceeded according to plan in 1991. The primary objective of the research was to scale-up the domains of application and demonstrate how well the learning methods work, how robust they are, and whether or not they require modifications and extension.

At the start of the project, Carbonell proposed to use a 60-operator metal machining domain, in which planning consisted of producing process plans to produce a metal part of given shape, size and features given raw materials and machines (milling, lathe, drill, etc.). They have subsequently expanded this domain into an over-100-operator domain whose range corresponds to that of an expert human machinist. Learning in this domain means solving problems more efficiently and producing more accurate plans. Learning from experimentation focuses on accuracy improvement (EBL and analogy research, not pursued under NASA sponsorship, address the efficiency issues).

1992 Progress

Carbonell's first objective was to grow and debug the domain. This proved not as difficult as expected, and the task is now complete. The second objective was to test out the experimentation method to increase the planning accuracy within this domain, and if necessary improve the experiment formulation module. The first part of this objective is complete: experimentation has been proven to scale up well with increased domain complexity: roughly linear-time increase in run time and comparable quality in newly acquired knowledge as in smaller domains. The second part of the objective requires a re-examination of the learning method to determine if it can be improved further. This will be the first task for 1992.

Mitchell and Mason have used a hand-eye testbed to explore learning in manipulation tasks. The basic task involves tilting a tray so as to slide an object to a desired location. Despite its apparent simplicity, this task has proven challenging to the robotics community, making learning an attractive approach to acquiring the necessary domain knowledge. Mitchell and Mason have constructed a number of simple learning agents that operate in this testbed. Experiments have shown that these agents can learn effective manipulation strategies that produce 95% success rates at positioning objects at randomly selected target positions and orientations. Furthermore, such learning techniques are effective across a variety of irregular object shapes and in the presence of complicating factors, such as additional invisible objects in the tray. These results represent one of the first robotic tasks for which learning techniques significantly outperform state-of-the-art analytical planning methods.

In the last year, Mason and Mitchell have extended this approach to handle continuous representations of the physical world. Learning occurs in the context of a back-chaining planner. For a given goal, and any hypothesized action, a pre-condition is formed by induction over past tilting operations. These pre-conditions then form the sub-goals for back-chaining. In experiments the system has achieved 96% success rates, demonstrating that effective learning is possible without requiring prior discretization of the state space.

1993 Plans

Carbonell plans to extend the PRODIGY planner to handle multiple probabilistic outcomes to operators, and explore the implications of learning by experimentation in a non-deterministic environment. Explore one of the NASA domains (actually start this in 1992) such as the photometry telescope planning/scheduling task or the shuttle attitude-jet planning, and determine the planner and/or learning components would apply and what benefits they may provide

Mason and Mitchell will develop and evaluate new agents for the tray-tilting domain that employ different amounts of background knowledge in the learning process, testing them across a wider range of situations than in previous experiments. One promising approach would take advantage of a naive theory of physics to constrain hypotheses about objects' behaviors. Second, they have begun exploring a different task domain, involving reinforcement learning of tightly integrated sensor-effector behaviors. Both the tray-tilting domain and the integrated sensor-effector strategies domain are designed to explore learning of purposeful behavior in physical tasks. The primary contribution is development of learning in continuous spaces, in the presence of substantial noise, with broad relevance to planetary exploration and operations in space.

Milestones

1992: - Release of the learning-by-experimentation module to external laboratories .
       - Produce a complete document describing all of the different aspects of learning
         by experimentation (Yolanda Gil's thesis).
1993: - Develop learning methods for manipulation that incorporate a
         naive physical theory to constrain hypotheses.
       - Initial exploration of integrated sensory-effector strategies.
1994: - Completely integrate the different learning methods
         of PRODIGY (experimentation, EBL, analogy, abstraction, static, KA) via a
         master-deliberator module,
       - Situate the integrated learning in a complex planning domain hopefully one
         selected from NASA or DARPA (a co-sponsor).
       - Continued exploration and evaluation of methods incorporating a naive physical
         theory.
       - Select the best learning system from previous studies with the hand-eye
         testbed and begin more detailed evaluation on complex manipulation tasks.

Learning to Use Devices
Paul Rosenbloom; USC/Information Sciences Institute

1992 Progress

Our research is focused on how an intelligent system can learn to use a new device, given that it has experience with related devices, a manual, and the device (or simulation) to experiment with. During the past year we have focused on three main topics: (1) finding an appropriate device to learn to use, (2) constructing a performance system for the device, and (3) hooking knowledge sources up to the system. These topics are treated in turn below.

On finding a device, we evaluated three NASA devices for appropriateness: the Shuttle's Reaction Control System (RCS) and Orbital Maneuvering System (OMS), plus the DataComm telemetry recorder system. Unfortunately, none of the three turned out to be appropriate, as they are all already highly automated, and what has been left to human intervention is either trivial or beyond the scope of our intended efforts. Given this resulting lack of an appropriate NASA device, we were forced to resort to a fallback device. Since domain proficiency was to be hand-coded into the system so that we could concentrate on the issues involved in learning devices, we were looking for a device with a fairly straightforward domain that was reasonably challenging to use. Since the video-cassette recorder (VCR) fits these criteria well, and has proved to be an excellent research vehicle, we are using it to drive our research until a NASA device can be found that also meet these criteria (even if not so well).

The purpose of constructing a performance system for the device is to provide a concrete embodiment of what the learning system is to produce. Our strategy is to start with a complete performance system for a very simple task -- showing a movie that is on a tape -- and then to back off a step at a time as we understand how to learn the various parts of it. Once the relevant portions of the system can be learned, we will move up to the next most complex performance task, and so on.

We started by determining which models would be needed for the "play a movie" task, how they would interact, and finding suitable representations for the models. We developed a number of systems during this time, each one solving the problems of its predecessors and unearthing new problems of its own. The current incarnation is fairly stable, and successfully solves the "play a movie" task. We began with three models -- device, domain, and mapping. However, this proved to be too limiting. In particular, we had trouble distinguishing between actions that could be performed deliberately by the system (pushing buttons) and the actions of the device that occur automatically in response to them. Handling automatic and intentional operators appropriately was, for a number of reasons, somewhat difficult. The problem was finally solved by adding a fourth model -- a user model -- plus making a number of representational changes.

Interaction between the models also provided a number of challenges. Although the basic approach varied little from the beginning -- a model utilizes one of the other models for guidance whenever it can not solve the task at hand -- the utilization structure between the models underwent several evolutions, driven primarily by consideration of the interactions between performance and learning. The current system has the domain space at the top being guided by the user space. The user space is in turn guided by the device space. The domain space unfortunately still learns some incorrect rules, but the user and device spaces learn correctly. We are still working out the details of the domain-space problems.

In developing this framework, we first concentrated on the device, domain, and user models, leaving the more complicated mapping step until later. This was managed by a few hard-coded rules in the models. Once these basic models were interacting properly, we removed these rules and added a primitive mapping space. This space takes a state in one model (space) and converts it into an equivalent state in another model by finding equivalents for each object in the

state. Of course, not all of the objects have equivalents, so the maps are often incomplete. We will begin to look at a more advanced schemes in 1992 (see below).

The first step in acquiring the models in the performance system is to develop the knowledge sources from which this learning will occur. The second step is to hook them up to the system so that they can be used when the performance system runs into trouble (that is, after some of its hard-wired knowledge is removed). To date, we have concentrated on a first pass through acquisition from similar devices (e.g., CD player, tape deck) and the manual. We added acquisition entry points to the general framework. This included methods for determining when the model's (problem space's) knowledge was insufficient for solving the problem at hand and thus needed to acquire new knowledge; for selecting an appropriate knowledge source; and for acquiring the knowledge via problem solving, mapping spaces and learning. (We have also begun to look at how a fragment of the manual's text could be understood via Soar's natural language capabilities.) Except for the first, these methods are still rather primitive and somewhat ad-hoc, but they have shown that the general framework is sound.

1993 Plan

During 1993 we will focus on the issue of acquiring models from multiple knowledge sources. At least at first, we will work only with the simplest performance scenario -- of playing a tape on a VCR -- and develop versions of the three basic knowledge sources, plus learning strategies, that can be used to acquire a model of the VCR sufficient to perform this task.

One of the key intellectual issues we expect to be addressing is how to perform the coupling between two arbitrary problem spaces that are used within a goal-subgoal relationship; that is, how to encode an impasse in one space into a problem to be solved in the other space, and how to encode the results of the solution in the subspace back into useful information in the parent space. (This problem can be viewed as a generalization of the analogy problem: in analogy, the two spaces are both device models, though about different devices; whereas here, in addition to this normal analogical relationship, there may be a need to map from a model of the domain to a model of the device, or between manual knowledge and a device space.) To date in Soar this issue has always been addressed by hardcoding translation rules that create initial states in new problem spaces and return results to parent spaces. However, such an approach is inadequate when the range of impasses that can occur, and the range of spaces that can potentially be used for them, is quite large. A general framework for performing such mappings -- even if not a general algorithm -- should go a long way towards providing the more flexible use of knowledge that is required in this task, and in many others.

During 1993 we will also continue to look for a second domain that is of direct relevance to NASA. One strategy we will consider is utilizing one of the devices investigated above, but with the current automated controllers removed (or limited). This would investigate how these controllers could be automatically acquired from the available knowledge sources.

Milestones

1992: - Develop a system that can acquire a VCR performance model (device and mapping spaces) from the three knowledge sources for at least the "play a tape" scenario.
- Develop a framework for mapping states and goals between models.
- Settle on an appropriate NASA device.
1993: - Expand the VCR system to deal with more complex scenarios.
- Implement first version of system to deal with simple scenarios for the selected NASA device.
1994: - Complete sy: em that can perform and learn over a range of scenarios for the NASA device.
- Experimentally analyze and evaluate the two systems along both functional and cognitive dimensions.

Improving Search-Based AI Systems
Thomas G. Dietterich; Oregon State University

The goal of this project is to develop scalable methods for improving the efficiency of problem solving. The focus is on optimization problems (i.e., those that occur in engineering design, scheduling, and minimax search). The main approach is to apply abstraction and learning techniques to convert general-purpose optimization algorithms into application-specific algorithms that are much more efficient and more reliable. The inefficiency and unreliability of existing general-purpose optimization methods is a major roadblock to the widespread adoption of optimal design methods in industrial and engineering domains.

1992 Progress

The project has studied three different optimization tasks. The first task is two-dimensional structural optimization in engineering design. Here we have obtained a speedup of more than 400% on small problems while improving the likelihood of finding an optimal solution. We expect even better speedups as the problem size scales. The speedups are obtained through a combination of three techniques: (a) an abstraction, called the "stress state", is introduced, which partitions the design space into disjoint regions, (b) search control heuristics are learned that decide--by examining the particular problem instance--which of these regions are likely to contain the optimal design, (c) partial evaluation methods are applied to convert the general-purpose objective function (which is a non-differentiable, non-linear function) into a special-purpose function (which is differential) by incorporating constraints from the problem instance and the specific region. Finally, gradient-based search procedures can be applied to this special-purpose objective function to search the selected regions of design space to find the optimal solution.

The stress state abstraction was discovered by graduate student Giuseppe Cerbone. Each stress state corresponds to a family of designs in which the forces in the members have specified signs, but unspecified magnitudes. To learn the search control rules, a new learning algorithm, UTILITYID3, was developed and tested. It must manage the tradeoff between efficiency of search (i.e., by selecting a small number of regions to search) and optimality of the solution. This tradeoff is represented by a utility function, and UTILITYID3 searches for a set of search-control rules that maximize this utility.

The second optimization task we have explored is the task of finding minimum-length schedules for space-shuttle cargo operations. We are following Zweben's formulation of this task as local search with a MOVE operator that moves an activity (along with its temporal dependents) from one point in the schedule to another point. Rather than applying this MOVE operator via simulated annealing, we are attempting to learn an evaluation function len*(MOVE(s)) that can predict the ultimate length of the schedule after applying the operator. We first tried to learn this function directly using the regression-tree algorithm of Breiman, et al., but of course this is quite difficult. To make the learning task easier, we are now focussing on learning an evaluation function len2(HC(s)), where HC is an operator that applies a simple hill-climbing algorithm to construct a satisfactory schedule. In a sense, len2 gives an estimate of the "height" of the "hill" that we are currently sitting on. We can then consider, for each possible MOVE, whether it will take us to a different, higher hill. In this way, len2 can be used as an evaluation function.

In fact, because len2 is represented as a regression tree, it is possible to analyze the tree to find the MOVE operator that is most likely to increase the value of len2. This is potentially much faster than generating-and-testing MOVE operators. All of this code has been implemented and tested. We are now in the process of conducting experiments to see if len2 can be learned accurately.

The third optimization task is chess. Graduate student Nick Flann completed his doctoral dissertation this year. His goal was to develop a learning technique that could acquire correct and optimal problem-solving knowledge, while possibly sacrificing completeness. He developed two forms of abstraction: (a) aggregating operators according to the goals they achieve and (b) aggregating board positions by geometric abstractions (such as rectangles, open lines, etc.). He applies these abstractions to learn optimal rules of play for chess endgames. The abstractions are

applied to reduce the cost of performing the dynamic-programming reverse-enumeration of endgames. In 1991, Flann tested his system on two chess endgames (King-Rook vs. King-Knight, and King-Rook vs. King). He was able to reduce millions of alternative moves into a few thousand generalized board positions, and thus obtain a speedup of several orders of magnitude over serial search. He also generalized the technique and tested it in the domain of checkers. Flann is now Assistant Professor in the Department of Computer Science at Utah State University.

Milestones

1992: - Test the performance of the 2D structural design methods on larger problem instances.
   - Develop better features for representing 2D design problems to improve generalization.
   - Develop features that could generalize across topologies.
   - Build a library of real-world 2D design problems.
   - Gather additional training data for predicting the len2 length of schedules.
   - Develop better features for representing schedules for learning.
   - Test the scheduling methods.

Soar and the External Environment
John E. Laird; University of Michigan

Soar is a general cognitive architecture that integrates problem solving and learning. The system represents knowledge as production rules that specify preferences for certain states, operators, or goals, and it uses this knowledge to direct search through various problem spaces. Before making a decision about which state to expand or operator to apply, Soar's rules add elaborations and preferences to working memory, which it then uses to select a state or operator. The architecture acquires knowledge through "chunking," a form of explanation-based learning that caches the results of previously solved problems.

Professor Laird's research at the University of Michigan uses Soar as a framework within which to develop an intelligent agent that interacts with an external environment. Over the last four years, the Soar group at Michigan has built systems within Soar for controlling a Puma robot arm and a mobile robot.

One of their thrusts during 1991 was to improve the environments used for experimentation. During 1991 they developed high-quality simulation environments for both the Hero and Puma robots. These are based on the IGRIP system that allows full three-dimensional, dynamic simulations. These simulations maintain sufficient fidelity so that it is possible to transfer software between the real systems and the simulations. In fact, both the Hero and Puma Soar software ran unchanged in the simulations the first time they were tried. There are many important advantages of these simulated environments: 1. Greater availability. The simulations can be duplicated by merely copying the software and do not require the actual hardware. Laird et. al. are sharing the Hero simulators with Johnson Space Center and Tom Mitchell's group at CMU. 2. Greater control of the environment. It is much easier to replicate experiments and automatically try out various configurations when testing the system. 3. Greater variety of environments.

1992 Progress

Within these environments, research during the last year has concentrated on developing and integrating many of the capabilities required for planning and execution in autonomous systems. These include coordinating multiple tasks that arise dynamically in an efficient and practical manner. This involves dynamically merging tasks so that tasks can be achieved in parallel, even though they may have originally been specified as independent sequential tasks. In the previous year we have developed a general facility in Soar for interleaving the execution of multiple tasks without requiring replanning. An extensive empirical investigation was done to verify the usefulness of the process in the blocks world, followed by an investigation of more complex and dynamic tasks implemented within the NASA TileWorld. Another capability being investigated is the management of limited dynamic resources, such as time, or fuel during the planning and

execution of tasks. Most current approaches assume perfect prior knowledge as to the costs associated with a given activity. Laird's group has developed a system that learns from experience the expected costs, and then adjusts these with experience as the environment changes. After sufficient learning, the system is able to dynamically modify its execution of tasks to meet the current resource constraints, even if they change unexpectedly.

1993 Plan

Laird's group has attempted to generalize its work on learning from errors that arise during interaction with an environment. The aim of this project is to define and implement a general weak method for error recovery by an intelligent agent. This involves detecting that an error has occurred in the agent's attempt to achieve its goals, determining a successful method for recovering from that error, and then correcting its knowledge in an effort to avoid the error in the future. This approach will then be combined with previous work on concept acquisition to enable generalization with problems that the agent has experienced previously. Previous work in on this project has emphasized the architectural mechanisms required to detect and correct errors. The emphasis has now shifted to exploring alternative methods for gathering whatever knowledge is available for aiding the recovery process. For example, they are studying the problem of learning by autonomously exploring and experimenting with a new task. Once these methods are all in place it will be interesting to see how much initial knowledge can be removed from the system with the agent instead learning from the errors it makes.

A completely integrated agent must be able to take advice, be instructable, and be able to perceive and act in the external world. Laird's group has produced an initial system which takes simple imperative commands in a high-level language and carries out the desired behavior in a simple control domain. There are a number of interesting research issues this work will address. For instance, it can be seen as a method for extending incomplete domain theories in explanation-based learning, whereas many approaches to this problem rely on inductive techniques, our system will ``learn by being told". However, the advice the system receives will be situated - related to the current situation at hand - so the system will have one example available to generalize from, guided by the advice. Since the system must learn from single cases, using an explanation-based approach, the breadth and depth of the initial domain theory should have a large effect on what can be learned. We plan to investigate how differences in the initial domain theory affect what can be learned from advice.

The most important aspect of Laird's research program is its emphasis on integration. Each of the subprojects will be tested in the same domains, encouraging the development of a unified knowledge base that can be used to control an integrated intelligent agent. This is a long-term goal, but the issues of attention, interruption, operator acquisition, and time-dependent behavior all lie along the path toward this goal.

Milestones

1992: - Augment Soar to support integration of multiple tasks.
- Finalize the work including in Soar an internal model of time, and test the resulting system on tasks which involve alternative actions that vary in the time they take to execute and in their reliability.
- Implement the resulting methods in both Hero-Soar and Robo-Soar.
- Augment Soar with the ability to take a variety of instructions.
1993: - Adapt the attention model as basis for a Soar sensor system that can develop information gathering routines for new goal-directed tasks.
- Augment Soar to learn new domain theory during experimentation.
- Develop an integrated hand-eye system based on Soar, and begin evaluation of the system.

Automated Abduction and Theory Refinement
Raymond J. Mooney; University of Texas, Austin, Texas

This project focuses on reasoning and learning in the presence of partial knowledge. Most work on automated reasoning assumes that complete logical proofs can be constructed for desired conclusions. Similarly, most work in explanation-based learning relies on the presence of a complete domain theory, and does not address the problem of revising an imperfect knowledge base. Raymond Mooney's work at the University of Texas is attempting to move beyond these limitations.

1992 Progress

In response to the first issue, Mooney and his students are developing an abductive reasoning system, ACCEL, which is allowed to make assumptions in order to complete an explanation of a set of observations. The initial system was developed during 1989 through 1991. During the past year, the focus was on improving the efficiency and capabilities of the system and evaluating it in several domains. The efficiency of ACCEL was improved by incorporating an ATMS (Assumption-Based Truth Maintenance System). Experiments were conducted showing that the caching of partial explanations provided by the ATMS decreased processing time by an order of magnitude on a range of abductive problems in plan recognition and device diagnosis. Methods were also developed that allow the system to retract assumptions and rules when subsequent observations contradict one of its current explanations. This allows the system to modify the underlying knowledge base as well as assumptions in order to accommodate new information.

Recent research has focused on applying ACCEL to a range of problems in automated diagnosis. First, the system was used to determine the areas of a medical patient's brain that were damaged during a stroke. Results on 50 actual cases with a knowledge base of 648 rules showed that ACCEL could efficiently diagnose cases as accurately as the specialized set-covering methods previously applied to this problem. Methods were also developed for performing abductive diagnosis of continuous dynamic systems modeled by a set of qualitative differential equations (QDE's). Future plans include using ACCEL to diagnose problems with the shuttle's Reaction Control System (RCS), for which a QDE model has already been developed.

Progress has also continued on Mooney's work on automatic refinement of incorrect and incomplete knowledge bases. An initial system, EITHER, was developed during 1989-1991 for revising propositional rule bases. During 1992, this system was completed and evaluated in two real-world domains. First, EITHER was extended to handle multiple category theories, constructive induction of new features, and noisy data. Next, it was tested in two domains: recognizing promoters in DNA sequences, and diagnosing soybean diseases. Using training sets of 100 preclassified examples, EITHER improved the accuracy of each of the initial theories approximately 30 percentage points. In both cases, EITHER was significantly more accurate at classifying novel examples than a standard inductive learning system (ID3) given the same training examples.

Since the expressive power of propositional logic is limited, recent research has focused on developing a new system, FORTE, for revising theories expressed in first-order predicate logic. This system can be viewed as automatically debugging PROLOG programs using sample input-output pairs. FORTE builds on previous developments in relational learning, specifically, Quinlan's FOIL system and Muggleton's inverse resolution methods. In 1992, an initial implementation was built and successfully tested on an imperfect theory for family relationships and the DNA theory used to test EITHER. A second implementation is now close to completion. The new version is more efficient and can revise and learn recursive theories.

Finally, methods were developed that permit FORTE to build and revise qualitative models of physical devices from behavioral data. Previous approaches to inducing qualitative models required that sample behaviors contain all relevant variables. Application of a general first-order induction system nicely solves the problem of introducing new variables into a model. Future

plans include using FORTE to perform diagnosis and model refinement of continuous dynamic systems, such as the shuttle RCS. Diagnosis can be treated as theory revision as follows:

1) A correct model of the device is given as an initial theory.
2) Behaviors of a faulty device are provided as examples that are inconsistent with this theory.
3) The system modifies the theory to account for the faulty behaviors.
4) The modifications are interpreted as faults in the device.

If the initial model is incorrect but the behaviors are accurate, then the same process is performed but the output is interpreted as an improved model of the device. If the initial model is empty, then the system creates one from scratch that accurately models the behavior. Therefore, a theory revision system can perform model formation and model revision as well as diagnosis.

Milestones

1992: - Complete extensions to ACCEL and FORTE to handle complexities of
        working with qualitative models of dynamic continuous devices.
      - Initial application of both systems to fault diagnosis in the
      shuttle RCS.

Representation In Incremental Learning
Paul Utgoff; University of Massachusetts, Amherst

This work consists of two projects, one on automatic model selection in inductive learning, and the other on learning from preferences exhibited by an expert. The first explores issues in selecting an appropriate representation, while the second focuses on issues in acquisition of expertise.

Automatic Model Selection:

The work on automatic model selection is motivated by the observation that no one learning algorithm is best for every learning task. Most algorithms obtain superior performance on some tasks, but this leaves us with the problem that one must try many algorithms in order to find which does best empirically. Each algorithm searches its own space of hypotheses for one that it identifies as the best rule (classifier) for identifying objects. One would like an automatic method that searches at least the larger hypothesis space implied by the collection of learning algorithms.

1992 Progress

The project has developed methods toward this end. Specifically, our LMDT algorithm builds a tree-structured hypothesis, with the test at each decision node being a linear machine (multi-class perceptron) that is itself induced. In this way, one combines the decision tree model with the perceptron model. During 1991, we developed the ability to discard irrelevant variables from a linear machine in an efficient manner. Since a univariate test at a node is a special case of a multivariate test, and given our ability to discard irrelevant variables, we intend LMDT to subsume some of the existing decision tree algorithms. Empirical evidence indicates that we have succeeded to a large extent, though the variable discarding mechanism still requires further refinement. The approach also subsumes learning with a single linear machine, since that too is a special case of the tree structure, and the tree structure allows one to learn tasks that are too difficult for a single linear machine.

The work is now proceeding with a more comprehensive approach to searching the hypothesis space, guided by heuristics that recognize characteristics of models and hypothesis pathology, and decide dynamically which model to try next. We intend to employ classification tasks of interest to NASA during this next phase, which will take approximately two years.

Automatic Expertise Acquisition:

The work on automatic expertise acquisition is motivated by the observation that people seem to be able to acquire expertise simply by watching an expert in action. This suggests that one ought to be able to automate expertise acquisition, and avoid the expensive process of extracting and engineering human expertise.

1992 Progress

The project has succeeded in developing a method for learning from the expert's choices by constructing pairs of state preference. For each action the expert could have done next, but did not, one infers that the resulting state from the selected action is preferable to the state that would have resulted from the action not selected. A novel method converts each such pair of states to a specific set of adjustments to the coefficient vector of an evaluation function. Thus, it is possible to learn an evaluation function, from state preferences alone, that will assign the highest value to the most preferred state. We also showed that this method complements the already known "temporal difference" approach to evaluation function learning, and that the two methods can be used individually to update the same single coefficient vector, leading to demonstrably faster learning than with either method alone.

During Summer 1991, we demonstrated the approach for air traffic control, assuming a rich but finite set of options available to a controller at any moment. By learning the controller's preferences, the system changes the order in which it displays options to the controller. Over time, one would expect the system to present the controller's preferred choice at the top of the list.

During Fall 1991, we demonstrated, on a dynamic system control task, that one can augment temporal difference learning with input from an expert to great advantage. For a much studied control task, only ten keystrokes were required by the expert in order to guide the temporal difference method in such a way that it learned the task much faster. This is important because reinforcement learning methods such as temporal difference learning are very slow.

For Winter 1992, we will integrate a method for learning to predict the resulting state, from a given state and operator, so that the state preference method will have even more applications. We would like to demonstrate our method on the space-docking task. Following that, we intend to tackle a challenge task involving more control inputs, such as showing that we can teach a machine to land an airplane. We are well positioned to solve such a problem.

Note, as described in the previous section, this activity is scheduled to end in 1992. It may be renewed as part of a competitive machine learning grants process.

Milestones

1992: - Implement heuristic method for searching model space (Brodley Ph.D. thesis).
- Identify NASA application, e.g., landsat analysis.
- Implement method for space docking task.
- Design and implement system for airplane landing task. (Clouse Ph.D. thesis)

Concept Formation in Problem Solving and Data Analysis
Douglas Fisher, Department of Computer Science, Vanderbilt University, Nashville, TN

Our research has continued to develop concept learning systems for three major application areas: exploratory data analysis, diagnosis, and planning. At the core, of much of this research, has been an unsupervised learning system called Cobweb, which clusters observations into similarity classes. During clustering, the system can use the evolving categorization structure to make predictions about any of an observation's missing features. Results in many domains

indicate that Cobweb quickly converges on highly predictive classification structures. We are now applying our basic methods to space-related applications, including interactions with researchers at the Jet Propulsion Laboratory and the Boeing Corporation.

1992 Progress

Our work in data analysis has proceeded in several directions. First, we are developing a more exploratory approach to clustering; traditionally, clustering systems uncover one primary categorical structure in data (e.g., mammals, birds, reptiles), even though others may exist (e.g., carnivores, herbivores, omnivores). A previous approach to this problem (i.e., creating graph-structured classification schemes) did not yield ideal results, but we have discovered a method that transforms an existing classification tree so as to bring out alternative class structures. A second line of research is motivated by the observation that in many domains there are attributes that one wishes to predict (e.g., design properties) given knowledge of others (e.g., specification properties). We have modified the evaluation measure used for clustering so that clusters are discovered that promote this type of selected inference. A special case occurs when it is desirable to predict values along a single dimension given knowledge of others. This is the task of supervised learning from examples. Thus, we believe that a novel and important contribution of this work is that it unifies two major paradigms of learning: unsupervised learning where no attributes are identified as uniquely important, and supervised learning where such identifications are made. This method is applicable to data analysis tasks using astronomical data at the Jet Propulsion Laboratory. One major task in this project is to identify types of astronomical bodies based on inherent properties and their relation to other objects.

A second major direction of our research has been to apply our concept learning systems to problem solving. Earlier work clustered math problem solution traces. New problems were then categorized using features given in the problem statement (surface features) to find best matching past problems; the solutions of these past problems were then modified and reused, thus reducing solution costs for the new problem. We have extended this work by exploiting 'derived' features in the categorization process. These derived features are not immediately perceivable in the problem statement, but derivable through inference with a domain theory. Identification of appropriate derived features improves problem solving efficiency over our previous methods, which used only surface features to categorize new problems. This research contributes to research on explanation-based learning concerned with identifying appropriate 'operational' predicates, and also explains data on human problem solving.

Currently, we are extending these methods to diagnosis and design-for-diagnosability. In particular, we have used clustering to discover fault modes in the heater loop of the (simulated) Space Station Water Recovery System. This process was used primarily to discover similar-looking fault conditions, thus suggesting sensor placements that would best disambiguate modes. We are also using these methods for diagnosis. Clusters describe probe points that best discriminate fault modes. In fact, we have developed a formal measure that orders probes so as to provably minimize the expected cost of diagnosis for variable-cost probes. This measure is used to minimize cost of an AND/OR search. Thus, it has implications for organizing domain theories used in explanation-based learning and decision (OR) trees. We are using a procedure in conjunction with this measure for diagnosis with one type of AND/OR structure called a fault tree. Currently, we are experimenting with simulations of a nuclear power plant, but we have obtained a NASA Ames model of the Reaction Control System of the Space Shuttle with plans to adapt our methods to this environment.

Finally, we are using clustering to organize plans into similarity classes, based on shared PRE and POST conditions. Our initial work on the Plot system clustered individual operators for means-ends planning, which led to improvements in planning efficiency. This year we have experimented with a learning planner that incrementally adds macro-operators to the abstraction hierarchy created through planning. Performance degrades due to predictable utility problems: macro-operators contain many more PRE and POST conditions (relative to individual operators) that must be matched during classification. Additionally, many macro-operators are not widely applicable to new situations, particularly under noisy conditions. These operators are best

removed from the abstraction hierarchy, and thus removed from consideration during planning. Our work on planning seeks to mitigate these problems by building in certain background knowledge that helps identify the most `relevant' features of operator definitions, and identifying low utility macro-operators that are best removed from consideration during planning. Through this basic research we hope to develop methods for planning in novel domains, such as space construction and planetary exploration.

Milestones

1992: - Apply data analysis tools to astronomical data at JPL, and possibly other sites.
- Integrate clustering and cost-minimizing diagnosis procedures into a power plant simulator and partially integrate into a model of the RCS.
- Mitigate utility problems in our planning system and conduct systematic experiments in noisy domains.

PHYSICAL SYSTEMS

The Design Reuse Assistant Project
Catherine Baudin, Jody Gevins
Larry Leifer, Ade Mabogunje, Venad Baya; Stanford Center for Design Research

Dedal, the Design Reuse Assistant, is part of the Design Knowledge Capture project whose long term goal is to develop methodologies for creating and maintaining an accurate record of a system design throughout its lifecycle. In this context, the goal of the Design Reuse Assistant is to facilitate the capture and reuse of previous design experience. The project is a collaboration between FIA and Stanford 's Center for Design Research (CDR) and is focused on capturing mechanical engineering design records. Dedal is a hybrid system which integrates multimedia design knowledge, such as formal artifact models, electronic engineering notebook pages, design meeting summaries and tech reports. Dedal uses device models for two purposes:

1. to point out the parts of a device which relate to a given requirement or to predict the impact of a design change. The system is coupled with a constraint-based CAD tool which manages the relationships among the design elements.

2. to act as an expert librarian which helps the user locate relevant segments of information in the Electronic Design Notebook text and graphics documents.

During the last two years:

- Stanford has investigated the use of tools and methodologies to electronically capture text and graphic records of the information usually recorded in designers' engineering notebooks. In particular, they studied the use of an Electronic Design Notebook to capture notebook pages, design meeting summaries and technical reports. The Electronic Design Notebook (EDN ™) is a hypertext system which includes facilities to manipulate text and graphics information, browse through the electronic documents and access external programs.

- In the meantime, our group at NASA Ames focused on capturing structural and functional artifact models. In particular we investigated ways of representing the relationship between the structure, and behavior of a device, and the initial design requirements. Capturing these dependency relationships is crucial to enable design modification and reuse, and to capture the rationale for design decisions.

1992 Progress

This year we focused our efforts on the following tasks:

1. Conducting a pilot study to identify classes of information that must be recorded to enable design reuse.
2. Studying how to represent the design information to facilitate design adaptation.
3. Testing the Electronic Design Notebook with a NASA designer.

• Software Implementation:

Dedal integrates the EDN with an engineering knowledge base which contains models of the artifacts being designed. We built an interface to our existing constraint-based CAD tool to facilitate the acquisition of the required engineering knowledge. We developed an intelligent design assistant module that takes as input a question from the user and uses its engineering knowledge to answer the question. Our reasoning component uses quantitative and qualitative knowledge about the dependencies among design parameters to predict the impact of a design change on the design requirements and decisions. We also developed an indexing scheme which uses *semi-formal* representations of the content of EDN documents to act as an intelligent indexing system. This helps the user locate relevant EDN design documents.

• Experimental studies:

We conducted two pilot studies to understand the information seeking behavior of a designer in the context of a design adaptation task. We asked two mechanical engineers, a Stanford graduate student and a NASA design engineer, to design a shock absorber by modifying a previously designed shock absorber. This shock absorber, the continuously variable damper, was designed at the Center for Design Research. Each engineer (subject) was given access to one of the experts who designed and documented the original damper. We extracted the questions asked by each subject from the verbal protocols recorded during these experiments. From these questions we identified classes of design knowledge that should be captured to enable design reuse.

• Representation and use of "machine intelligible" device and requirement models:

The Stanford (CDR) mechanical engineer who designed the damper spent three months at NASA Ames. During this time he:

1. used the object oriented language of Dedal to model the structure of the damper, the design requirements, and the relationships between these requirements and a set of design choices. We used this model to generate explanations for design rationale and to implement design aids by simulating the impact of a design change on the requirements and constraints.

2. used the model and the design vocabulary generated from our protocol studies to describe the content and the form of the shock-absorber documentation in the EDN. These descriptions define a set of indexing patterns and are built by combining an element of the structural damper model with an element of the design vocabulary previously identified.

We collaborated with Jean-Charles Bonnet from the Stanford Knowledge System Laboratory (KSL) to implement a behavioral model of the damper using the qualitative simulation engine DME developed at KSL by Yumi Iwasaki and Tom Gruber. This model is not yet integrated with Dedal but it helped us understand the advantages of using a qualitative simulator to predict the effects of a design change on a device behavior.

• Evaluation:

We conducted an experiment to evaluate how Dedal addresses the following issues: use of engineering device models to help the redesign process, intelligent retrieval of EDN documents, and the ability of a user to formulate questions to the system.

1993 Plans

We will extend this research in the following ways:

NASA/Stanford:

- Testing: We will continue to test Dedal's retrieval performance and the ability of a designer to use Dedal's design vocabulary to formulate questions to the system. In our next experiment, a user will be directly interacting with Dedal to solve a new problem.

<u>NASA:</u>

- Interactive refinement of the artifact model: our testing with Dedal showed that the engineer who defines the artifact model (structural, functional, behavioral and requirement models) should be able to interactively modify the model while running the system on a set of sample problems. For instance, if the result of a simulation does not corroborate with the expectations of the designer, he should be able to easily modify the model until his expectations are met.

- Semi-automatic acquisition of indexing patterns: in this year's experiment, the semi-formal representation of our multimedia documents were entered manually by an engineer. We will study how to semi-automate this phase using a question-based indexing method which uses the questions asked by a user to extend the initial set of indexing patterns.

- Using comparative analysis to predict the impact of the structural changes on the device's behavior: in the current version of Dedal, the system can predict the impact of a change on a requirement but our model does not integrate the qualitative differential equations needed to deal with time dependent requirements. This extension will require us to pursue the integration of Dedal with DME or with theQSIM family of simulators.

- Extend the Computer Aided Design capability of the system by completing the integration of a symbolic algebraic manipulation package (REDUCE) to help the designer define and solve equations.

<u>Stanford:</u>

- Stanford will continue testing the knowledge-base developed to represent the Damper design.

- Study a new query interface for Dedal which would incorporate high-level language processing techniques to parse English-like questions.

- Conduct new design knowledge acquisition sessions with engineers using the Electronic Design Notebook instead of their paper-based engineering notebooks. Stanford will supervise the use of EDN to capture design information during the design of a Satellite, Satellite Test of the Equivalence Principle (STEP) project.

Milestones

1993: - Dedal user tests completed.
- Semi-automatic indexing of design information incorporated into VMACS.
- Pilot use of EDN in the Automation and Machine Design course at Stanford.
- Develop a natural language like query interface
- Create new knowledge bases for a NASA artifact
       STEP project data acquisition underway
   SS Centrifuge Facility project candidacy under study
- Continue testing the Electronic Design Notebook with NASA engineers.
1994: - Complete integration of EDN in STEP and one other project.
- EDN extended to include CAD/CAM tool access.
- Complete the development of an interactive tool to help modifying device models.
1995: - Full integration of Design Knowledge Capture tools with the Concurrent Engineering Enterprise tools being developed in the SHARE project.

Diagnosis of Physical Systems
Deepak Kulkarni, Peter Robinson
Matt Barry, JSC/Rockwell MCC
Ben Kuipers, Herbert Kay; University of Texas at Austin

The diagnosis and monitoring of complex, engineered systems is one of the most prevalent technical problems posed by NASA missions. NASA invests a major portion of its budget on

training and maintaining a cadre of personnel capable of monitoring and diagnosing missions. These people include ground controllers at mission operations and space flight operations centers, payload specialists at payload operations centers, and astronauts in the Space Shuttle.

Recently, first generation expert systems have been successfully applied to diagnosis and monitoring in a variety of systems including RTDS (JSC), SHARP(JPL), and PI-in-a-Box. These expert systems are *associational* because they associate symptoms with diagnostic hypotheses. They are also *experiential* because their knowledge is based on previous experience with the systems themselves, as well as experience gained from testbeds and simulations. Our research focusses on *model-based* systems that differ from associational or experiential systems because their knowledge is based upon fundamental models of the structure and behavior of engineered systems. They base their diagnoses on the discrepencies between expected behavior (derived from a simulation of the model) and observed behavior. Model-based systems are generally dependency-directed where discrepencies are backtraced from their manifestation until the underlying cause of the problem is found, or they search between multiple models of the system trying to match fault models against the observed behavior until a model that explains the observed behavior is found.

This research element is intended to strengthen the Ames research program in advanced methods of diagnosis and to forge a transition pathway back into NASA operational missions. To accomplish this we have focussed upon Space Shuttle subsystems to guide our research and are developing a facility that will be a direct datalink to the JSC Mission Control Center (MCC). This mini-MCC will enable Ames personnel to collect and log data from actual Shuttle flights and from large-scale Shuttle simulations. This data will be made available to researchers in the machine learning and diagnostic communities to encourage research relevant to NASA missions.

1992 Progress

Space Shuttle RCS   Monitoring and Training Assistant
Deepak Kulkarni, Peter Robinson
Herbert Kay; University of Texas at Austin

MCC mission controllers face a number of problems in diagnosing a system with failed sensors or degraded components. First, standard malfunction procedures can not be executed in the presence of failed sensors. Secondly, it is difficult to entertain and track a large number of hypotheses that could be consistent with data over time. Finally, some failure patterns of the systems, unreported in training manuals, are needed for inferring the state of the system. To assist Reaction Control System (RCS) mission controllers in overcoming these problems, we are developing an advanced fault diagnosis assistant software (SRFD) in collaboration with Matt Barry, an experienced mission controller at JSC. In addition to its use in mission operations, SRFD will also be useful in training the shuttle mission controllers. At present, mission controllers are trained using traditional lectures and hands-on simulation. Mission controllers do not get feedback during hours long simulations as over-the-shoulder tutoring by teachers is expensive. SRFD will serve as an intelligent feedback system in training mission controllers by providing analysis of the current state of the system when they need it. This will, in turn, improve the quality of available training to mission controllers.

SRFD is based on Dvorak and Kuipers' MIMIC framework for this problem. MIMIC has four components: a hypothesis generator, a model-constructor, a model-simulator and a matcher. The hypothesis generator examines current readings of sensors to generate a set of hypotheses. The model constructor generates models from these hypotheses. The simulator predicts future behaviors of the system. The matcher validates hypotheses by testing the predicted behaviors against real data. The hypothesis generator postulates faults using model-based reasoning, learnt rules from real data, and expert-defined rules. It can accept incomplete current state description as in the case of failed sensors. The simulator uses a qualitative QSIM model of the RCS which can handle nominal modes such as toggling isolation valves or firing thrusters. In addition a variety of fault models such as failed open or closed regulators, helium tank and leg leaks, and thruster leaks can be modeled which allows it to simulate the system even when one or more components

have degraded. Finally, the matcher is responsible for comparing actual data against predicted behaviors in order to calculate its confidence in each hypothesis. In summary, the system is able entertain possibly large number of hypotheses and then narrow them down to a single one, if further data rules out the rest. Its use of model-based reasoning, learned failure patterns and qualitative reasoning gives it the ability to analyze degraded systems well.

In 1992, we developed an initial version of SRFD and demonstrated its ability to diagnose multiple failures in regulator system by testing it on JSC shuttle simulator data. Our initial efforts focussed on producing a tractable QSIM model of RCS for the simulator component of the system.. We found that a QSIM model produces a very large number states when we explicitly model each tank and leg in the each RCS. We proposed an abstraction of components as a possible solution. We showed that the regulators can be modeled as an abstract single system without loss of any information. However, when the multiple fuel paths from tank to thrusters are modelled as a single path, there is a loss of information. In 1992, we developed an understanding of proper use of abstractions in monitoring and diagnosis. In 1993, we propose to solve this problem by integrating quantitative information with the qualitative RCS model. We also found that MIMIC does well on idealized data, but on shuttle simulator data, we need to develop schemes to handle noise. In 1992, we developed and tested a number of methods to handle noise in model-based reasoning and qualitative reasoning.

1993 Plan

We plan in 1993 to extend the hypothesis generator to include rules learned using machine learning algorithms and expert-defined transition rules. We also plan to integrate quantitative information with our qualitative RCS model. In addition, we will extend SRFD to capture further capabilities of the RCS system. We will also use probabilistic methods to assign confidence by integrating these in simulation and MBR. In 1994, we will carry out large scale tests of the system to qualify it for mission use. Later in 1994 and in 1995, we plan to integrate SRFD with intelligent electronic RCS manuals, and with PRS, a system that suggests procedures.

In 1992, as reported in Knowledge Acquisition and Use During Design Section , we developed models of the Reaction Control System (RCS) of the Space Shuttle in support of diagnosis research at Ames. This domain and these models are also driving work at the Stanford KSL, the University of Texas-Austin, and at Vanderbilt University. We hope to widely disseminate these models and encourage multiple modeling efforts. This work is being performed in collaboration with Matt Barry, who is an experienced, Shuttle RCS mission controller . As modeling and diagnosis research matures, and is validated within the Ames mini-MCC, we expect to work with Matt to transition systems back into the MCC. We also hope to collaborate with the JSC team developing the Space Station Control Center (SSCC) to possibly integrate some of our model based techniques into their "computation builder " and "display builder ".

Self Calibrating Models for Dynamic Monitoring and Diagnosis
Ben Kuipers, Herbert Kay; University of Texas at Austin

The goal of this project is to develop and demonstrate a method for automatically building qualitative and semi-quantitative models of dynamic systems, and using them for monitoring and fault diagnosis. Qualitative modeling and simulation are particularly useful in situations such as fault diagnosis, that require reliable reasoning with incomplete knowledge. QSIM makes it possible to express and reason with a qualitative description that may be reliably known, in contrast with a numerical model that could require assumptions of functional form and numerical parameter values. Quantitative information may be expressed as bounds on landmarks and other symbolic elements of the qualitative description, by adaptively inserting new time-points to improve the resolution of the description and converge to a numerical function, and by deriving envelopes bounding the possible trajectories of the system. Both qualitative and quantitative inference are designed to preserve soundness: all behaviors consistent with the model are predicted. This reliable modeling and simulation capability is the foundation for the MIMIC approach to monitoring and diagnosis of dynamic systems. In MIMIC, multiple hypothesized models of the system are tracked in parallel in order to reduce the "missing model" problem. Each

model begins as a qualitative model, and is unified with *a priori* quantitative knowledge and with the stream of incoming observational data. When the model/data unification yields a contradiction, the model is refuted. When there is no contradiction, the predictions of the model are progressively strengthened, for use in procedure planning and differential diagnosis.

We plan to demonstrate how self-calibrating, numerically bounded models can be acquired and used for monitoring continuous systems. A "numerically bounded model" is a qualitative model with numerical bounds on landmarks, monotonic functions, and predicted behaviors. It preserves the basic QSIM soundness guarantee that the predicted behaviors cover all real possibilities.

"Structural bounds" are bounds on landmarks, envelopes on monotonic functions, and bounds on certain other symbolic elements (e.g., distances between landmarks and slopes of monotonic functions) of the QDE and initial state. "Behavioral bounds" are bounds or envelopes on the predicted behaviors. (Q2, Q3, and NSIM are three methods for inferring behavioral bounds from a qualitative model and its structural bounds.) A "self-calibrating model" is a qualitative model with a learning procedure capable of acquiring the structural bounds from a stream of observations of the behavior of the physical system.
A MIMIC-based monitoring system includes several distinct types of inference:

•**diagnosis**: initially, or given some event, propose one or more plausible qualitative models to explain the state of the system;
•**self-calibration**: unify the model with observations to converge on a numerically bounded model with minimal uncertainty;
•**prediction**: use the numerically bounded model to predict unobserved present or future behavior of the system;
•**monitoring**: manage the set of models and data streams to maintain and track a tractable set of alternate possible models of the system.

In the case of the Space Shuttle Reaction Control System (RCS), the automatic generation of numerically bounded models supports several mission-critical functions:

• Fault hypotheses (e.g., leak, jet fail on, jet fail off) can be identified and at least partially evaluated at an earlier point in the mission, before RCS shutdown permits isolation tests.

• A numerically bounded model of a system with one fault (e.g. a small leak), increases the system's ability to detect the occurrence of a second fault in the same system.

• In a highly reconfigurable system such as the RCS, the ability to fit a self-calibrating model to a new configuration should support both monitoring and the planning and evaluation of new operating procedures.

• While normal operating procedures specify that a leaking segment of the RCS is not to be used, under mission-critical circumstances it may be necessary to use a portion of the system in spite of a significant leak. In such a case, the availability of a numerically bounded model of the faulty system could make the difference between success and failure of the mission.

Currently, numerical models are developed by the designers of the system and its operating procedures, and by ground controllers in response to emergency situations. By automating these methods and making them part of the normal monitoring system, we hope to increase the availability of these models and their adaptation to the specific configuration and state of the device.

This project will deliver the following.

•General-purpose tools, in the form of software extensions to the QSIM system, providing the relevant capability. Early versions of these tools will be provided to collaborators at ARC and elsewhere for rapid feedback.

•Scientific papers describing the advances underlying these tools, for publication in AAAI, IJCAI, AIJ, etc.

•Domain models drawn from the RCS or other suitable subsystems of the Space Shuttle, demonstrating the performance of the tool and its contribution to mission success.

Milestones:

1992: - Implement a numerically bounded modelling system and ensure that the qualitative-quantitative simulation algorithms are usable and efficient.
- Construct a qualitative model of the RCS fuel and oxidant resource usage and demonstrate the ability to converge on a quantitatively bounded model of that system by assimilating a stream of observations.
- Collaborate with the Kulkarni-Robinson project to integrate this capability into their monitoring system.
- Develop an abstracted QSIM model of RCS.
- Demonstrate the ability of SRFD to diagnose multiple failure regulator faults by testing it on JSC shuttle simulator data.
- Develop schemes to handle noise in model-based and qualitative reasoning.
1993: - Extend the range of the self-calibrating model method to more complex systems.
- Implement diagnostic algorithms for retrieving or constructing plausible hypotheses given unexpected events in the observational data.
- Demonstrate the use of multiple-time-scale models and multiple-time-scale data using the Cheung & Stephanopoulos method.
1993: - Integrate quantitative information with qualitative RCS model.
- Extend the RCS model to capture further capabilities of the RCS system.
- Develop sophisticated scheme of evaluating confidence in hypotheses.
1994: - Demonstrate the use of abductive learning methods (e.g. MISQ and FORTE) to propose qualitative models by analyzing observations.
- Apply the self-calibration methods to converge those qualitative models down to quantitative models of previously unknown systems.
- Test the system on shuttle simulator and mission data, and qualify it for mission use.
- Integrate SRFD with PRS to provide extended FDIR capabilities.
1995: - Integrate SRFD with intelligent books.
- Continue integration of SRFD with PRS.

SIGMA: Scientists' Intelligent Graphical Modeling Assistant
Richard Keller, Michal Rimon, Aseem Das, David Thompson
Michael Sims; ARC, Computational Systems Research Branch
Chris McKay; Ames Theoretical Studies Branch
Jennifer Dungan; Ames Ecosystem Science and Technology Branch

The goal of this project is to provide computational support for scientists engaged in computer modeling and simulation of physical systems, such as planetary atmospheres, forest ecosystems, and biochemical systems. Although computer models play a crucial role in the conduct of science today, scientists lack adequate software engineering tools to facilitate the construction, maintenance, and reuse of modeling software. Usually, scientific models are implemented using a general-purpose computer programming language, such as Fortran. Because this type of general-purpose language is not specifically customized for scientific modeling problems, the scientist is forced to translate scientific constructs into general-purpose programming constructs. This manual translation process can be very complicated, labor-intensive, and error-prone. Furthermore, the translation process obfuscates the original scientific intent behind the model, and buries important assumptions in the program code that should remain explicit. The resulting software is typically complex, idiosyncratic, and difficult for anyone but the primary scientist to understand.

We are building a knowledge-based software environment that makes it easier for scientists to construct, modify, and share scientific models. The SIGMA (Scientists' Intelligent Graphical Modeling Assistant) system functions as an intelligent assistant to the scientist. Rather than construct models using a conventional programming language, scientists can use SIGMA's graphical interface to "program" visually using a more natural high-level data flow modeling language. The terms in this modeling language involve scientific concepts (e.g., physical quantities, scientific equations, and datasets) rather than general programming concepts (e.g., arrays, loops, counters). SIGMA assists the scientist during the model-building process and checks the model for consistency and coherency as it is being constructed. Then SIGMA automatically translates the conceptual model into an executable program, freeing the scientists from error-prone implementation details. To provide this level of automation, the system must be given a significant amount of knowledge about the scientific domain, as well as general knowledge about programming. This knowledge is represented and stored in a large knowledge base that contains information about scientific equations, physical quantities and constants, scientific units, numerical programming methods, and scientific domain concepts and relations. In constructing this tool, we are using a variety of advanced software techniques, including knowledge representation, automatic programming, and constraint propagation techniques from artificial intelligence, as well as techniques from object-oriented programming, graphical interface design, and program visualization.

To ground our work, we have been conducting a case study of a planetary atmospheres model developed (in Fortran) by collaborator C. McKay. His model uses scientific data from Voyager I's flyby of Titan to infer the thermal and radiative structure of Titan's atmosphere. Our methodology has been to "reverse-engineer" the Fortran model in order to reconstruct the knowledge and methods used by the scientist to build such a model. Based on our analysis, we are developing a general tool that will ultimately enable atmospheric scientists to construct an entire class of atmospheric models -- including McKay's Titan model -- using our automated modeling assistant. To avoid developing a system that is too narrowly scoped, we also have been studying models in other scientific domains. In this way, we increase confidence that our efforts are general and transferable to other scientific disciplines.

1992 Progress

During 1992, we made progress on three key fronts: (1) we increased the range and complexity of modeling programs that SIGMA can generate; (2) we significantly enlarged the scope of the system's knowledge base by expanding into a second scientific domain; (3) we began work on a graphical interface to support user interactions with SIGMA.

Improvements in SIGMA's program generation capabilities include its ability to handle structured data (such as arrays), to numerically integrate certain types of ordinary differential equations, and to make program calls to external subroutines provided by the scientist. As a result of these enhancements, we were able to use SIGMA to reconstruct the entire "atmospheric setup" section of McKay's Titan model in just a fraction of the time that would have been possible without automated assistance.

Also during this year, we expanded beyond planetary sciences into an earth sciences domain. In particular, we extended our knowledge base to support modeling activities associated with the Forest-BGC ecosystem model. This forest carbon and water-cycle model was developed by Dr. Steven Running at the University of Montana, and is used by collaborator J. Dungan and other researchers in the Ames Ecosystem Science and Technology Branch. Dungan's goal is to use SIGMA to build a generalized version of Forest-BGC that can be used to investigate regional-scale forest ecosystem phenomena.

Our third major accomplishment during 1992 was the design and partial development of a new graphical user interface for SIGMA. The interface is a critical component of the system because it facilitates complex information flow between the scientist-user and the system. The new interface, when complete, will enable users to flexibly construct and manipulate data flow models. The

previous SIGMA interface forced users to develop models using a constrained and sometimes awkward form of model-building interaction.

1993 Plans

For 1993, our plans build upon work in the three focal areas described above. To further enhance program generation capabilities, we plan to extend the basic graphical programming language by introducing conditional data flow constructs, as well as various forms of iteration. We also plan to introduce hierarchical grouping constructs as a means of dealing with model complexity. To implement the generalized Forest-BGC model, we will introduce methods for handling temporal simulation, which are necessary to reimplement the Forest-BGC model. During 1993 we also plan to complete our new base-level graphical user interface and implement extensions to permit users to enter and edit elements of SIGMA's knowledge base, including scientific equations. Finally, during 1993, we will attempt to broaden our base of users by extending beyond our primary collaborators in planetary atmospheres and ecosystem sciences. Longer term, we will make our tool available to atmospheric scientists supporting the Cassini mission and the Huygens probe to Titan, as well as to global change scientists analyzing EOS instrument data.

Milestones

1992: - Add conditional data flow, iteration, convergence constructs to data flow language.
       - Enable hierarchical data flow constructs
       - Implement Forest-BGC temporal simulation
       - Complete base-level graphical user interface, and deliver to Ames Code SGE & SST collaborators
       - Build graphical equation editor
       - Implement cloud and haze code sections of Titan model
1993: - Complete balance of Titan model, including radiative transfer section
       - Write major journal article describing SIGMA
       - Extend numerical and symbol-processing capabilities of system by utilizing external packages
       - Enable user editing of domain theory objects and relations
1994: - Port system to Macintosh environment to facilitate wider distribution to scientific community
       - Expand into a life-sciences domain

DTA-GC Instrument Control System
David E. Thompson, Richard Levinson, Peter Robinson
Rocco Mancinelli, Lisa White; ARC Code SSS

1992 Progress

This project is developing software that autonomously controls a new biogeochemistry laboratory instrument and that provides assistance in the chemical and structural analysis of soils and their constituents. The new instrument combines two well-known materials analysis methods: differential thermal analysis (DTA) and gas chromatography (GC). Although the coupling of these analysis systems results in a more detailed characterization of the minerals and organics in the soil samples than has previously been available, their combined use also has required the development of new reasoning expertise, detailing how the data or results of the two types of analyses interrelate. This expertise has been gained through the systematic construction during this year of the integrated DTA-GC instrument itself, through development of the control and reasoning software in synchrony with this construction, and finally from the use of the system on soils and mixtures whose chemical decompositions provide clear examples of the interplay between thermophysical and chemical processes. Furthermore, we have previously explained that the approach for building the interactive software has been first to build a system capable of controlling an experiment run, then of reasoning about the full set of (possibly partially) matched data in order to suggest next-case experimental protocol needed to resolve ambiguities in

identification. This "serial" capability of the software is what is now being completed. The transition to operations in an interrupt mode will be designed this year but will be developed and tested in 1993. That mode exploiting parallel reasoning and execution, whereby the system carries out partial matches of data with library records while the data is "coming in" from a run, and thus allows re-programming of an experiment profile during that run, based on expectations. As the response software evolves into a real-time planning and control system, there will be renewed effort towards encoding general operations procedures, experiment design, monitoring and control procedures, and procedures for fault isolation and recovery.

In order to fully transition into the parallel operations mode in 1993, and to document emerging experience and plan for future extensions of the software and hardware, a "white paper" will be written that explicitly defines performance criteria and evaluation standards for all the software. The criteria will be different for serial and parallel operations, thus allowing operational comparisons, and they will be enhanced as more working knowledge is gained. The procedures used for these standards and criteria will also be encoded, yielding performance metrics which can guide our extensions and help empirically determine the meaning of 'improvements' to the system. However, we will also describe the transition of the DTA-GC project into an Autonomous Laboratory focus (AutoLab). The flexibility which has been implemented in the Sparcstation makes it fully capable of controlling multiple experiments at a time, thus carrying out DTA-GC experiments while controlling and reasoning about other (possibly unrelated) analytic experiments. We see the development of AutoLab not only as a strongly enhancing tool for the NASA experimental scientist, but also as a step towards the future development of an autonomous platform for reactive science experiments, capable of executing multiple experiments and analyses in remote or hostile environments. A major opportunity to expose this emerging technology is the recent formal invitation extended to Dr. Mancinelli to be co-investigator with the Germans and Russians on the planned Mars '96 Mission, in which Mancinelli would provide scientific mission rationale using the DTA-GC. Hence, as this opportunity develops, our team intends to analyze and pursue whatever contributions can be made. If the autormtated DTA-GC is accepted as a mission critical element, we will investigate a teaming arrangemnent with an aerospace contractor (or an equivalent NASA internal group) to flight-qualify our system. We hope to add more personnel in 1993 for the purpose of aiding in the engineering of new systems (both s/w and h/w), and perhaps in system diagnosis research.

1993 Plans

Previous reports concerning plans and directions of this project have suggested that the team hoped to extend the capabilities of the DTA-GC system itself by the addition of more scientific tools, such as an XRD for verification crystallographic integrity, or an infrared spectrometer for capturing information about grain size or adsorbed surface species, and so on. While many of these types of tools do enhance an analytic laboratory, they are expensive, and are best added due to a discovered need from failure of the current system. We have instead concluded that the best extension of our system is in a related analysis domain, but which does not require additional physical connections to be made to the DTA-GC. To this end, it is our team's intention to design both the hardware and software interface for a Chemostat/Bioreactor. The construction of the system and the porting of the generalized DTA-GC software into that system will occur in 1993.

A chemostat is a device used to monitor and control both the population density and growth rate of a microbial culture. Generally, three elements are used in its control: (1) the flow rate of nutrients into the culture; (2) the flow rate of of waste products out of the culture; and (3) the concentration of a limiting nutrient, such as carbon, nitrogen, or energy. For example, at low concentrations of an essential nutrient, the nutrient is quickly used up by microbial growth, but in the chemostat continuous addition of fresh growth medium containing the limiting nutrient permits continued and regulated growth. To maintain a microbial culture, the level of the limiting nutrient, as well as all other growth parameters (temperature, pH, O_2 levels) must be monitored and corrected as necessary. The current "state of the art" in controlling chemostat type devices is equivalent to the level of control of the commercially available DTA software: they maintain a preset inflow/outflow rate, temperature, pH, and nutrient input, but real time automated decision

making is totally lacking in such systems. We propose construction of our own chemostat from purchase of the appropriate pumps plus machining labor, and our own software development.

Chemostats are useful in genetic, physiological and ecological studies with micro-organisms. In ecological research, chemostats can be used as model systems to study microbial growth in nature under conditions of different nutrient limitations or diurnal growth patterns. Chemostat vessels can be linked together in a series to form multistage chemostats, incorporating spatial and temporal heterogeneity into a study. Such a system is ideal for studying sequential metabolic activities performed by different microbes, in soil, sediment, or aquatic ecosystems. This type of ecological modeling research is of interest to NASA programs in ecosystem science (e.g., effects of global change on nutrient cycles), exobiology (e.g., the effects of a planet's physico-chemical environment on early evolution of life), and controlled ecological life support systems (CELSS). On a broader scale, development of an expert system to monitor and control a chemostat-type device has obvious industrial applications in the biotechnology, chemical engineering, pharmaceutical, brewery or fermentation, and food industries.

We believe that a coupling of a DTA-GC and a chemostat in that DTA-GC would be a powerful scientific instrument. The DTA-GC provides analysis of natural soil-site environments for microbial cultures, and the chemostat helps dictate the environmental conditions necessary for the observed organic signatures in the DTA-GC data. Hence, the environment modeller for DTA-GC is made far more robust, and process chemistry analysis tools can be developed for both systems in concert.

During 1994, the chemostat system and DTA-GC will together be generalized as the initial development of AutoLab, and the two will provide our next step towards the Autonomous Platform for Reactive Science Experiments.

Milestones

1992: - Complete testing of integrated DTA-GC in vacuum.
       - Completion of enhanced data analysis system and serial operations controller.
       - White paper defining performance criteria and evolution to AutoLab.
       - Design of chemostat/bioreactor and s/w interface for control and reasoning.
1993: - Analyze and pursue Mars 1996 contributions.
       - Construction of Chemostat; s/w interface built.
       - Generalization of DTA-GC s/w for the Chemostat.
       - Extend DTA-GC classifier from mineral to soil/rock modeller and to geologic environment modeller; extend system to parallel operations.
       - Development of analytic process chemistry tools for DTA-GC and Chemostat.
1994: - Generalization of s/w from DTA-GC and from Chemostat to AutoLab concept.
       - White paper on transition from AutoLab to Autonomous Platform for Reactive Science Experiments.
       - Continue appropriate Mars '96 contributions.

Superfluid Helium On-Orbit Transfer (SHOOT) Flight Project
Eric Raymond, Jeff Shapiro, Gary Villere

This project focuses on the development of software tools to support the monitoring and control of a Shuttle experiment called SHOOT: Superfluid Helium On-Orbit Transfer. Ames' role in this joint effort with the Goddard Space Flight Center is to provide all software used to control and monitor the experiment during all operational phases. Highlights of this effort include the first use of an expert system in space, an innovative use of laptop computer technology, a highly interactive, user friendly system used to teleoperate the payload from a remote ground station, and a simulation system used as both a development and training tool.

The SHOOT experiment is a Shuttle payload bay cryogenics experiment scheduled to fly on STS-54 in April of 1993 on the orbiter Atlantis. The SHOOT experiment consists of two insulated containers (dewars) of liquid helium connected by a transfer line. The helium is normally in a state called "superfluid" in which it appears to flow without viscosity. This superfluid component of helium has effectively no entropy and display quantum mechanic behavior at bulk, macroscopic level. The main experiment objective is to demonstrate the ability to transfer this odd liquid from dewar to dewar, much as would be done during a cryogenic servicing of a helium cooled orbiting telescope such as the planned Space Infrared Telescope Facility (SIRTF).

SHOOT will determine if all this is technically possible. Software being developed will provide facilities for the control and monitoring of the experiment as each hardware element and procedure are exercised. The software effort consists of three major components:

- AFDeX - An autonomous control, monitoring, and diagnostic system
  operated by the Shuttle crew to perform transfer operations.

- CMS - An interactive, user friendly Command and Monitoring System
  operated from the ground by a highly trained operator.

- CATS - A Command and Telemetry Simulator which emulates the payload for
  developmental testing of software and crew training.

AFDeX (pronounced /af-deck/) is a rule based expert system written in the CLIPS (a forward chaining production system language developed by NASA) and C languages. The system operates on a flight qualified GRiD 1530 80386-based laptop computer called the Payload and General Support Computer (PGSC) and will be located on the Shuttle's Aft Flight Deck (AFD). The system is designed to control the experiment without any support from ground or crew; it is completely "in the loop". The system interprets telemetry from the payload with respect to its current goals and sends commands to the payload in order to achieve these goals. If a problem occurs in the payload, AFDeX will attempt to diagnose the situation and recover. (Diagnosis is performed via empirical associations.) A user may optionally interact with the system to modify its behavior. The intent is to provide an autonomous control capability which is tolerant of a variety of failures and able to incorporate external knowledge from a user (perhaps as the result of earlier experiments in the mission) at execution time. Additionally this software provides the crew with real-time feedback from the payload which is used to coordinate a number of experiments involving thrusting of Reaction Control System (RCS) jets.

In contrast to AFDeX, the CMS is a ground based system for payload operation and control which provides an expert user with a highly configurable, fine grained interface with the experiment. While AFDeX contains expert knowledge tailored for specific phases of the experiment, the CMS is general purpose tool which is used in all phases of operation. This system along with the CATS simulator represent the current state-of-the-art in event oriented, mouse controlled user interfaces of the Macintosh computer.

Other responsibilities of the SHOOT software staff include supporting payload qualification testing, operation of the payload at Kennedy during post shipment checkout and integration, and operation of the payload from the Goddard Payload Operation and Control Center during flight.

Milestones:

1991: - Flight dewar tests. Flight-like operations under control of the CMS and
       AFDeX software. Simulation of payload via CATS simulator.
1992: - Astronaut crew training begins with AFDeX and the CATS payload simulator.
1993: - SHOOT is launched on STS-54 on the orbiter Atlantis.

PI-In-a-Box

Silvano Colombano, Michael Compton, Richard Frainier
Irv Statler, Ames Aerospace Human Factors Division
Larry Young, Nicolas Groleau, Lyman Hazelton, MIT

The goal of the Principal Investigator-in-a-Box project is to improve the scientific return of experiments performed in space by providing astronaut experimenters with an "intelligent assistant" that encapsulates much of the domain and experiment-related knowledge held by the principal investigator (PI) on the ground. By using expert systems technology and the availability of flight-qualified personal computers, it is possible to encode the requisite knowledge and make it available to astronauts as they perform experiments in space. The system performs four major functions: diagnosis and troubleshooting of experiment apparatus, data collection, protocol management, and detection of interesting data. The project team's efforts this year focused primarily on ground testing of the system in support of the first Spacelab Life Sciences Mission (SLS-1) in June of 1991. The system helped collect and analyze actual data from the "Rotating Dome", an experiment in vestibular physiology devised by Professor Laurence Young of MIT.

1992 Progress

The PI-in-a-Box system was used in support of the Rotating Dome experiment during three phases of the mission. During pre-flight data collection at the Baseline Data Collection Facility at JSC, the system was used to help collect data at launch-minus-150 days, launch-minus-75 days, launch-minus-45 days, launch-minus-30 days, and launch-minus-15 days. These data, which represent how the subjects perform under normal conditions in 1G, served as a baseline against which to compare the data collected in subsequent in-flight and post-flight sessions. During the actual mission, the Rotating Dome experiment was carried out on Mission Day 5 and Mission Day 6. During these experiment sessions, the PI-in-a-Box system was used on the ground near the Science Monitoring Area (SMA) at JSC. The system was connected to the same stream of raw data that was downlinked from the Spacelab and monitored by the PIs inside the SMA. After the mission, the system was used to collect post-flight data at NASA's Dryden Flight Research Facility at Edwards AFB, CA. These data collection sessions, which took place on return-plus-0 days, return-plus 1 day, return-plus-2 days, return-plus-4 days, return-plus-7 days, and return-plus-10 days, measured the astronauts' responses as they re-adapted to gravity.

The primary accomplishment of the PI-in-a-Box system in its testing during SLS-1 was that it proved that the system works under realistic conditions. The data collection and data quality monitoring modules correctly acquired and interpreted the data, and were able to keep up with the real-time data stream. The data analysis routines correctly interpreted the data and provided meaningful quick-look analyses and statistical summaries of the experiment data that were printed out and taken in to the PIs in the Science Monitoring Area, who generally agreed with the analyses. PI-in-a-Box generated new protocols that included steps to pursue the "interesting" data and that would make optimal use of the time remaining for the experiment. These new protocols were also printed out and taken into the SMA, where they were used by the PI team to plan for the possibility of subsequent experiment sessions.

After the SLS-1 mission, the project team focussed on refining the system based on our experience during ground testing. These refinements included reducing the frequency with which experiment protocols are recomputed, modifying the user interface to include "electronic strip-charts" that display the raw data being collected by the system, and tighter integration of the Diagnosis and Troubleshooting Module with the rest of the system.

Finally, we began to integrate the modules comprising the two-computer architecture onto a single computer. This integration will increase the robustness of the system as a whole as well as its usability in space.

1993 Plans

Continuing project efforts will focus primarily on preparation for flight of the system during the SLS-2 mission, currently scheduled for May, 1993. This will involve re-hosting of the system onto a flight-qualified Macintosh Powerbook 170 laptop computer.

Other space scientists have expressed interest in investigating the applicability of the PI-in-a-Box system to their experiments. These potential applications would serve as a means by which to help generalize the system and would result in a system that could be useful to a broader segment of the space science community. These potential applications include:

- **The Vestibular Sled Experiment**: Another of Professor Young's experiments, the Vestibular Sled measures human adaptation to weightlessness from a slightly different perspective than the Rotating Dome. The Sled experiment is planned for flight on the second International Microgravity Laboratory mission (IML-2) in the 1995-96 time frame.

- **Modeling of Planetary Atmospheres**: An experiment devised by Tom Scattergood and Chris McKay involves studying the formation of organic aerosols in Titan's atmosphere by simulating that environment in the so-called Gas Grain Simulation Facility.

- **Cell Growth in the Weissman Apparatus**: This experiment, devised by Rose Grymes, explores how the growth of lymphocytes is affected by microgravity.

- **Biomedical Monitoring and Space Research Centrifuge**: These domains, being investigated by Robert Mah, would involve monitoring of astronaut physiology and conditioning, and control of experiments that use the Space Research Centrifuge.

We will begin working with these investigators, as appropriate, to produce proposals for joint efforts which allow continuation of this project into some form of deployment on Space Station Freedom.

We also plan to begin integrating PI-in-a-Box with the Space Station Freedom Data Management System (DMS) testbed under development at Ames. Our first step will be to ensure compatibility of data transmission and retrieval between our system and the DMS.

Milestones:

1992: - Re-hosting of the system on the Macintosh Powerbook 170 laptop computer
       - Completion of flight hardware and software for SLS-2
       - Initiation of integration efforts of PI-in-a-Box with the Space Station DMS testbed
1993: - Pre- and post- flight support of SLS-2 during baseline data collection sessions
       - Flight of PI-in-a-Box system aboard SLS-2 (mission postponed from
         previous 8/92 date)
       - Generalization of system and application to another experiment
1994: - Continuation of generalization effort
       - Completion of integration with DMS testbed

Intelligent Purchase Request System
Michael Compton, Helen Stewart
Shirin Tabibzadeh; ARC, Information Sciences Division Computer Systems Group
Barbara Hastings; ARC, University Affairs Branch

The Ames Acquisition Division processes tens of thousands of purchase requests (PRs) per year. These PRs, which can be used for anything from procurement of computing equipment and laboratory instruments to allocation of funds for university grants, must be approved by anywhere

from five to twenty people before they can be processed. More than half of these PRs need to be returned to the originator because they are incorrect or incomplete.

In 1991, as a "weekends and evenings" experiment, the Ames AI Research Branch, University Affairs Branch (ASC), and Information Systems Branch (EDI) began development of a prototype system to handle a subset of PRs; namely, those that pertain to university grants that are managed by NASA. This initial prototype system, completed last year, consists of approximately six electronic forms that can be completed by individuals in NASA and submitted electronically for approval to people in the Information Sciences Division, Aerospace Systems Directorate, and the University Affairs Office. The prototype runs on Macintosh computers that are connected by local Appletalk networks and the Ames LAN, and is built with off-the-shelf software (SmartForm Assistant from Claris Software and CE Software's QuickMail).

1992 Progress

In 1991, the project team focussed on establishing the mechanisms and infrastructure necessary to permit the origination, submission, and approval of university grant PRs. This year we are extending the system to cover a wider variety of PRs; namely those that apply to "small purchases" (i.e. purchases up to $25,000). These small purchase PRs are considerably more complex than those that pertain to university grants, and will need to be approved by a larger and more diverse group of people. In order to adequately address this broader area, it is necessary to improve the system in two important ways; more stringent validation of the PR before submission and automatic generation of an "approval path" based on the content of the PR.

The original prototype system was designed to perform limited "local" validation on each form. That is, it ensured that numeric fields contain only numeric data, and that all required fields were filled in. We are now developing a facility for "global" validation of the PR and its associated documents. This facility will not only check that all required forms accompany a PR, but also that information across those forms is consistent and correct.

The form validation module, being developed in CLIPS, consists of form-specific sets of constraints that define acceptable values for fields contained on those forms, and a form-independent constraint-application mechanism that ensures that each of the constraints are met. The validation module also includes general input-output routines that read data that is extracted from the forms being validated and pass back error messages for each of the constraints that were violated. This architecture permits the extension of the validator to new forms by adding new constraints without modification of the constraint-application mechanism.

The approval-path generation module, also CLIPS-based, consists of potential "approvers" grouped by organization (such as "Division Office", "Directorate Office","ADP", "Safety", etc.). Each approver is characterized by a name, title, electronic mail address, and alternates. The module also includes a mechanism to determine which organizations must approve the PR (depending on its content) and will select the proper approver from each organization to which the PR must be sent. The module will then construct a sequence of approver email addresses which dictates the electronic path that the PR will take during approval.

Milestones:

1992: - Completion of the prototype Validation module and Approval-path Generator
        - Evaluation of the prototype system on a representative subset of small-purchase PRs

Knowledge Based Software Engineering
Michael Lowry, Smadar Kedar
Penny Nii, Stanford University

Knowledge-based software engineering (KBSE) integrates knowledge-based methods, formal methods, and conventional software engineering techniques. Software can be viewed as

knowledge that has been compiled from many different sources and then operationalized to run on a computer. With conventional software engineering techniques, this knowledge compilation process is labor intensive with little opportunity to reuse anything but the
final code. In the long run, the aim of KBSE is to elevate software engineering to the 'knowledge level' - that is, building on a library of domain knowledge and design knowledge, a human interactively develops a domain-oriented specification, which is then semi-automatically transformed into efficient code. Systems built with KBSE will be self-documenting and self-modifying: humans will be able to query such systems about all aspects of the knowledge that went into their development, and then incrementally change the specifications of these systems. Changing the specification will cause a KBSE-generated system to replay its development, appropriately modified according to the deltas in the specification.

These long-term capabilities will significantly facilitate building future generations of complex NASA software systems, particularly long-lived systems that undergo considerable evolution throughout their life span. In the short run, the aim of KBSE is to develop high-impact tools whose knowledge-based techniques improve software engineering productivity. The goal of the KBSE project at Ames is to pursue a research program addressing long-term capabilities while also developing short term applications. Dr. Michael Lowry was hired recently to lead the project. Lowry served as PI at the Kestrel Institute, an internationally known research organization specializing in KBSE. Lowry also recently edited a collection of papers on KBSE that is being published through MIT Press.

1992 Progress

During the past year, Lowry and Kedar collaborated in developing the framework for a possible near-term application: the Learning Translator. Translating programs between different languages and hardware platforms is a labor intensive maintenance task; for example, in 1990 JSC assigned forty programmers to the task of porting Fortran programs on IBM mainframes to C programs on UNIX workstations. Although commercial off-the-shelf translators are available, they are seldom used. Such translators treat the target language as a machine language, making the resulting code unreadable. It is also difficult to specialize these translators to different programming language dialects. Thus, although these translators satisfy the minimal functional goals of porting - - enabling code to be run on a new platform -- they do not satisfy the non-functional goals such as maintainability which are usually the primary motivation for porting code.

The Learning Translator assists a maintenance programmer who is porting a program by ensuring that functional goals are satisfied, such as target program correctness, while providing implementation freedom in meeting non- functional goals. The envisioned Learning Translator will provide a knowledge acquisition capability, so that the maintenance programmer can simply provide examples of how code fragments should be translated, and the system will generalize these examples into translation rules. After the translation rules are generated, they are then interactively applied in order to translate code. A prototype system that syntactically generalizes examples of translated code fragments, the Generic Translator, has already been implemented. In contrast, the Learning Translator would semantically generalize examples of translated code fragments, by deriving the preconditions under which a translation rule is functionally correct. The Learning Translator has been simulated on examples of translating data structure operations.

Penny Nii, at Stanford University, began development work on the KASE project - a knowledge-based software design environment focused on redesign - that is, the modification of an existing design to fit new requirements. KASE will provide support to users in assembling a logical architecture for a blackboard system from a library of design components, customizing the architecture to a particular application, and when requirements change redesigning the architecture. KASE extends the capabilities of current CASE tools, from merely documenting design decisions to actively assisting in the process of making design decisions.

1993 Plans

In the coming year, additional examples will be simulated for the Learning Translator, including control flow translations. Based on an assessment of the likely utility to NASA, a decision will be made on whether to proceed to a prototype implementation.

The main focus of 1993 will be development of the framework for a retrieval/adaptation assistant that will eventually be integrated with the JPL Encyclopedia of Software Components being implemented by Dr. Beckman at JPL. The long-term research will combine techniques from AI and formal methods, and complement the less formal techniques being developed by Dr. Beckman and Penny Nii at Stanford University. An outside grant will be established for research on application of formal methods to software reuse.

Given a user's specification of a desired software system, the retrieval/adaptation assistant will find the most specialized (and hence, most efficient) configuration of subroutines from a library that can verifiably achieve the specification. This will be done by automatically developing theory interpretations from the specifications of the subroutines to the user's specification. Those parts of the user's specification that cannot be covered will comprise a residue that will need to be coded by hand, or generated through automatic program synthesis. The retrieval/adaptation assistant will also derive reformulation functions for transforming the representations specified by the user to the internal representations specified for the subroutines (and also between subroutines). For example, the user might specify a sparse matrix representation, while a subroutine might require a full matrix representation because of its internal address computation, or vice versa. Eventually, the retrieval/adaptation assistant will be extended to specialize code in a library to a specification in order to enhance run-time efficiency.

A domain of NASA relevant software will be chosen, and then formal specifications will be developed for a library of mathematically oriented subroutines. The specifications for these subroutines will be encapsulated as theories and hierarchically indexed by specificity. The links in this hierarchy will be interpretations between theories. Methods will be developed for constructing theory interpretations from the specifications for subroutines to portions of a user's problem specification. The retrieval/adaptation assistant will be simulated with various problem specifications from the domain to validate the utility of these methods.

As part of the integration effort with the Encyclopedia, this subroutine library will be processed with a commercial tool for analyzing Fortran programs to produce standard CASE documentation, including data flow and set/use of variables. This documentation will then be ported to the hypermedia representation of the Encyclopedia. This code-level documentation will help users understand, verify, and adapt retrieved programs. The code-level documentation will complement the formal specifications developed as part of the research, and provide an evolutionary path toward the more advanced capabilities of the retrieval/adaptation assistant described above.

Milestones:

1992   - Develop framework for retrieval/adaptation assistant.
        - Chose a NASA relevant software domain, and develop formal specifications for a library
          of subroutines.
        - Develop methods for mapping specifications of subroutines to a user's problem
          specification.
        - Assess possible utility of Learning Translator, and extend from data structure operations
          to other constructs, such as control flow.
        - Interface code analysis to Encyclopedia.
1993   - Implement prototype for retrieval/adaptation assistant.
        - Choose a second NASA software domain to help guide development of
          retrieval/adaptation assistant.
1994   - Field retrieval/adaptation assistant with the Encyclopedia.

Approximate Reasoning-Based Intelligent Control
Hamid R. Berenji

As the complexity of a control system grows, the task of developing an exact mathematical model of the system becomes more difficult, and in many nonlinear systems it is impossible to develop such a model. The objective of this research is to provide methods in control of physical dynamic systems in which we can start with approximate control rules (e.g., obtained from human experts) and learn to refine this knowledge by automatic learning.

The approach selected here differs from the conventional control theory in that we use a heuristic model for controlling a system. Our goal is to develop methods for learning, refining, and improving the knowledge based control systems which use approximate reasoning or fuzzy logic. The research focuses on providing adaptive capabilities to the current non-adaptive techniques used in fuzzy logic control. We concentrate on reinforcement learning techniques which can operate without the presence of a supervisor during the learning process and, hence, are more applicable to control problems.

1992 Progress

We have developed a number of techniques for integrating approximate reasoning with reinforcement learning. The ARIC architecture can learn to refine a knowledge base of fuzzy control rules based on observing the performance of the controller during different trials. Briefly, ARIC consists of three main elements: the control action evaluator, the control action selector, and a control action modifier. The task of the action selector is to recommend a control action based on a knowledge base of approximate control rules. The control action evaluator then evaluates this action and the state of the system. Based on the action selected and its evaluation, the action modifier makes final adjustments and sends the conclusion to the physical system. As a result of applying the adjusted control action, the state of the system changes into a new state and the above cycle is repeated. During this process, the rules in the knowledge base are updated in each cycle to improve the performance of the controller. The ARIC architecture when tested on benchmark control problems has shown orders of magnitude improvements in the learning speed when compared to neural network approaches for control.

We extended the ARIC architecture to G-ARIC which can model a more generalized class of fuzzy control rules and provides a full integration of a fuzzy inference system with reinforcement learning. To study the feasibility of the G-ARIC architecture for nonlinear control problems, we applied it to several small, but challenging control problems including cart-pole balancing and automatic backing up of a truck to a docking location. Two journal papers, describing the results of this research, were accepted for publication. Based on the superior results of these experiments over the previous methods published in the literature, we initiated experiments on using the G-ARIC's method for fuzzy adaptive control in the attitude control of the Space Shuttle. This project is a part of a larger project in automatic rendezvous and docking of the Space Shuttle with the Space Station Freedom. Our preliminary results showed successful attitude hold (e.g., pitch rate control) within a small region known as the deadband.

1993 Plans

Our plans center on three main topics: the continued development of the theory for automatic refinement of control systems based on approximate reasoning; application of our method in the obstacle avoidance and path planning of a simulated EVA system; and the continued application of our control method to the simulation of the rendezvous and docking operations. In addition to demonstrating the performance of these systems using software simulations, we plan to demonstrate their performances in a prototype laboratory system. This system consists of a two-dimensional micro-gravity table and two moving platforms on air-bearings which can float on the surface of this table using their on-board trusters.

Milestones:

1992: - Demonstrate the application of our adaptive fuzzy control scheme for attitude control of the Space Shuttle using STS docking simulation software obtained from JSC.
- Demonstrate the application of G-ARIC in obstacle avoidance and path planning on two-dimensional micro-gravity table .
- Extend the theory for the generalized aggregation operators in fuzzy logic control.
1993: - Complete and publish an analysis of NASA domain applicability of fuzzy methods.
- Demonstrate a complete integrated prototype of an operational fuzzy logic-based control system.
1994: - Complete and publish the theory for automatic refinement and learning of intelligent control systems which use approximate reasoning.

Fuzzy Control
Lotfi Zadeh; University of California at Berkeley

The major goal of our research is the development of computationally effective methods for the analysis and design of intelligent control and knowledge-based systems which operate in an environment of uncertainty and imprecision.

1992 Progress

In 1992, we have initiated new directions in our research which promise (a) to contribute to a better understanding of the design methodology for fuzzy rule-based systems; and (b) to lead to novel ways of designing neuro-fuzzy systems in which neural network techniques are employed to infer rules under supervised or unsupervised learning.

The first project is aimed at the development of what is called the *calculus of fuzzy if-then rules*. This calculus is a self-contained subset of fuzzy logic. Its importance stems from the fact that most of the current applications of fuzzy logic in the realm of intelligent control involve a manipulation of fuzzy if-then rules. However, the calculus of fuzzy if-then rules goes far beyond the standard rule-based techniques for representing qualitative input-output relations and using interpolation to compute the response to a given input.

The principal items on the agenda of the calculus of fuzzy if-then rules are the following: semantics of a single rule; semantics of a collection of rules; manipulation of blocks of rules; algebraic operations on collections of rules; interpolation; induction of rules from observations; applications to intelligent control, qualitative systems analysis and qualitative decision analysis; and self-organization, adaptation and learning.

In our approach, we consider three types of rules: *(i)* unqualified rules, e.g., Y is B if X is A; *(ii)* qualified rules, e.g., usually (Y is B if X is A); and *(iii)* rules with exceptions, e.g., Y is B if X is A unless X is E. Particular importance is attached to the semantics of rules with exceptions. This issue is closely related to default reasoning, non-monotonic reasoning and circumscription.

To deal with blocks of rules, we have employed FA-Prolog to represent the meaning of fuzzy if-then rules and implement their combination. To deal with algebraic operations on fuzzy if-then rules, we have developed procedures for computing the fuzzy if-then rules for a specified combination of collections of such rules. In particular, our approach makes it possible to compute the fuzzy if-then rule representation for the product of two functions h(X) and g(Y) from the knowledge of fuzzy if-then rule representations for h(X) and g(Y). This technique has an important application to computation with imprecisely known probabilities. More specifically, if the probability distributions p(X) and q(Y | X) are represented as collections of fuzzy if-then rules, then we can compute the corresponding representation for the joint probability distribution r(X,Y) = p(X) q(Y | X). In this way, we can compute fuzzy expectations as well as joint and marginal probability distributions.

1993 Plans

Our research has a direct bearing on the analysis and design of systems which may be employed directly or indirectly in space missions of various types. Experience with fuzzy-logic-based control systems has shown that the use of such systems enhances the robustness of system performance and reduces power requirement. Our development of the calculus of fuzzy if-then rules is aimed at further enhancement of the ability of fuzzy logic to improve system performance. We plan to continue the development of the techniques described above and, in addition, investigate the following problem areas:

- Construct and test algorithms for combining blocks of fuzzy if-then rules.
- Develop methods for representing knowledge expressed in a high-level, symbolic language as a collection of fuzzy if-then rules.
- Develop methods of imprecise matching based on the use of possibility and necessity measures, and employ such methods for improved approximation in interpolative reasoning.
- Develop the application of dynamic programming to the induction of rules from observation.
- Apply the calculus of fuzzy if-then rules to qualitative systems analysis and qualitative decision analysis.
- Develop methods of qualitative probabilistic reasoning using linguistic variables and FA-Prolog.

Milestones

1992: - Develop methods for converting qualified fuzzy if-then rules into unqualified fuzzy if-then rules.
- Develop methods for performing algebraic operations on blocks of fuzzy if-then rules.
- In collaboration with Dr. Hamid Berenji, develop applications of reinforcement techniques to intelligent control.
1993: - Develop inference techniques for qualified fuzzy if-then rules.
- Develop methods for computation with probability distributions represented as collections of fuzzy if-then rules.
1994: - Develop methods of induction of fuzzy if-then rules from sparse observations.
- Develop methods of induction of fuzzy if-then rules from databases.

Research on Multi-Use Knowledge Bases and Knowledge Interchange Technology

Edward A. Feigenbaum, Richard E. Fikes, Yumi Iwasaki, Tom Gruber; Knowledge Systems Laboratory, Stanford University

Research Objectives: To develop a methodology and a computational framework that will enable engineers to access a comprehensive and reusable body of engineering knowledge and reasoning methods for a wide variety of tasks, such as design verification, diagnosis, behavior analysis, and interactive documentation. To develop techniques for (a) building ontologies in a form that is translatable into the specialized representation languages of multiple application system environments, and (b) interchanging the reusable content of knowledge bases, including ontologies, among specialized representation languages.

Approach: Focus on three reasoning tasks: formulation of device models, integrated qualitative/quantitative simulation, and machine-generated explanation of device behavior. Also focus on representations for sharable, reusable knowledge bases, development of an engineering ontology, and reasoning with multiple models. Select a small number of interesting devices as drivers of the methodology. Build a prototype computational environment for device modeling.

1992 Progress

Constructed a knowledge base within the Device Modeling Environment (DME) to model the Reaction Control System (RCS) of the Space Shuttle. The knowledge source was NASA mission controller Matt Barry. The model of the RCS is being used to simulate execution of fault procedures. The expected behavior representation and the mechanisms for using the knowledge to detect desirable (or undesirable) consequences will also be useful for the purpose of operator procedure verification.

Expanded the DME knowledge base to support continuing work on modeling of the Hubble space telescope's Electrical Power System (EPS), including alternative component models (e.g., chemical properties vs electrical properties of batteries), component failure modes, and multiple levels of granularity.

Modeled components of a "planar" robot manipulator as part of an experiment in distributed simulation and design support being conducted by the Palo Alto Collaborative Testbed (PACT), which includes research groups at Stanford, Lockheed, Hewlett-Packard, and Enterprise Integration Technologies.

Developed a technique for automatically formulating device behavior models from libraries of component behavior models based on given modeling constraints and goals.

Incorporated modeling and simulating of discontinuous events and continuous changes into DME.

Ported the initial implementation of the system on CYC to EPIKIT, and optimized the system, resulting in a 20x improvement in performance.

Developed a framework for stating and reasoning about relevance and irrelevance of knowledge to enable selection of those parts of a large knowledge base that are most likely to be useful for a given goal.

Developed a facility for selecting the most appropriate perspective from which to model each component of a device depending on the problem context.

Developed a technique for validating that the approximation assumptions underlying a model enable prediction within the desired error tolerance.

Developed an explanation facility that makes use of model formulation and simulation traces of DME to generate explanations about the structure and behavior of a device. The technique is based on real engineering models, is scalable with model libraries, and is driven by interactive queries of several types.

Developed initial version of a language, called Ontolingua, for describing ontologies and technology for translating those descriptions into appropriate statements in multiple representation languages.

1993 Plans

In 1993 the research on multi-use knowledge bases will extend the functionality of the DME in several aspects. A special language will be designed and implemented to allow the user to enter a functional specification of a device. Similarly, a language for describing the rationale for particular design choices will also be implemented. Automatic model formulation based on context dependent behavior (Gruber and Nayak 1992) and on considerations of relevance (Levy and Iwasaki 1992) will be incorporated into the DME. Languages for modeling operator actions and procedures will be developed and implemented, and tested in the context of the RCS.

In the area of Knowledge Interchange Technology the plan for 1993 includes a revision of the KIF specification and demonstration of its utility as an interlingua for knowledge sharing among different representation systems. The initial KIF specification will be presented at major AI conferences and a summary paper will be published. We plan to demonstrate, in semi-automatic mode, translation into and out of a state-of-the-art representation language (e.g., CLASSIC -> KIF -> CLASSIC). Some existing engineering domain models will be examined to determine whether they are representable in KIF, and the results reported. We will also write a report evaluating KIF as an inter-agent communication language in the Stanford Bay Area community's agent-based engineering testbed (PACT). The initial KIF specification will be revised based on feedback from testing and evaluation.

Milestones

1993:   - Automatic context-based and relevance-based model formulation.
        - Automatic exploration of alternative simulation scenarios guided by user-provided constraints.
        - Incorporation of operator procedures into model formulation, simulation, and explanation methods.
        - Completion of revised KIF specification based on feedback from testing and evaluation.
        - Demonstration of semi-automatic translators sending a nontrivial knowledge base on a round trip from one representation system to another and back (e.g., CLASSIC -> KIF -> LOOM -> KIF -> CLASSIC)
1994:   - Automatic verification that design satisfies functional specification.
        - Design rationale acquisition facilities.
        - Context-based model formulation extended to include dynamic models, multiple operating regions, quantitative descriptions, and behaviors that   prevent or regulate processes.
1995:   - Automatic generation of additional constraints on a design and initial conditions that would guarantee that the device will satisfy the functional specification.
        - Use of design rationale to alert the designer to decisions that need reconsideration as a result of changes in design requirements, available alternatives, or operational environment.
        - Explanations of how requirements are reflected in a design and of function in terms of structure and behavior.

BIBLIOGRAPHY:
JOURNALS, BOOKS AND MAJOR PEER-REVIEWED CONFERENCES

Almuallim, H. and Dietterich, T. G. (1991). Learning with many irrelevant features. *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, 547-552. Anaheim, CA.

Bakiri, G. (1991). *Converting English text to speech: A machine learning approach.* Oregon State University. Corvalis, OR.

Berenji, H. R. (1992). *Fuzzy and neural control, in: An introduction to intelligent and autonomous control*, Antasaklis, P.J. and Passino, K. M. (Eds.). Elsevier, North-Holland.

Berenji, H. R. (1992). An architecture for designing fuzzy controllers using neural networks. *In the International Journal of Approximate Reasoning.*

Berenji, H. R. (1991). On integration of reinforcement learning and approximate reasoning. *In the 30th IEEE Conference on Decision and Control.* Brighton, England.

Berenji, H. R. and Khedkar, P. (1992). Learning and tuning fuzzy logic controllers through reinforcements. *To Appear in IEEE Transactions on Neural Networks.*

Bresina, J., Drummond, M. and Kedar, S. (1992). Integrated reactive systems pose new problems for machine learning. S. Minton and P. Langley (Eds.), *In Machine Learning Methods for Planning and Scheduling.*

Brodley, C. E. and Utgoff, P. E. (1992). Univariate vs. multivariate decision trees. *Submitted to the Ninth International Conference on Machine Learning.* Technical Report. Department of Computer Science, University of Massachusetts. Amherst, MA.

Buntine, W.L. (1991). Classifiers: A theoretical and empirical study. *In the International Joint Conference on Artificial Intelligence.* Sydney, Australia.

Buntine, W.L. (1991). Some properties of plausible reasoning. *In the Seventh Conference on Uncertainty in Artificial Intelligence.* Anaheim, CA.

Buntine, W.L. (1991). Theory refinement of bayesian networks. *In the Seventh Conference on Uncertainty in Artificial Intelligence.* Anaheim, CA.

Buntine, W.L. and Weigend, A. (1991). *Bayesian back-propagation.* Complex Systems, Vol. 5.

Buntine, W.L. (1992). *Learning classification trees.* D. Hand (Ed.), Artificial Intelligence and Statistics, Chapman and Hall. London, England.

Carbonell, J. G., Knoblock, C. and Minton S. (1990). *Prodigy: an integrated architecture for planning and learning, in architectures for intelligence.* K. VanLehn (Eds.), Lawrence Erlbaum.

Chapman, D. and Kaelbling, L. P. (1991). Input generalization in delayed reinforcement learning: an algorithm and performance comparisons. *Proceedings of the International Joint Conference Artificial Intelligence (IJCAI)*, pp. 726-731.

Cheeseman, P., Kanefsky, B. and Taylor, W. (1991). Where the really hard problems are. *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI)*, 331-337. Sydney, Australia.

Christiansen, A. D. (1991). Manipulation planning from empirical back projections. *Proceedings of the IEEE International Conference on Robotics and Automation*, 762-768.

Christiansen, A. D.,. Mason, M. T. and Mitchell, T. M. (1991). Learning reliable manipulation strategies without initial physica. *Robotics and Autonomous Systems*, Vol. 8, pp.7-18.

Christiansen, A. D. (1992). Learning to predict in uncertain continuous tasks. *Submitted to the International Conference on Machine Learning.*

Clouse, J. A. and Utgoff, P. E. (1992). A teaching method for reinforcement learning . *Submitted to the Ninth International Conference on Machine Learning.* Technical Report 92-7. Department of Computer Science, University of Massachusetts, Amherst. MA.

Covrigaru, A. A. and Lindsay, R. K. (1991). Deterministic autonomous systems. *AI Magazine, 12 (4),* pp.110-117.

Dietterich, T. G. and Bakini, G. (1991). A general method for improving multiclass inductive learning programs. *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91),* pp. 572-577. Anaheim, CA: AAAI Press.

Dietterich, T. G., Hild, H. and Bakiri, G. (1992). A comparison of ID3 and backpropagation for English text-to-speech mapping. *To Appear in Machine Learning.*

Dietterich, T. G. (1991). Knowledge compilation: Bridging the gap between specification and implementation. *IEEE Expert, Vol. 6, 2,* pp. 80-82.

Fisher, D. H., Pazzani, M. J. and Langley, P. (Eds.) (1991). *Concept formation: knowledge and experience in unsupervised learning.* San Mateo, CA: Morgan Kaufmann.

Fruchter, R., Iwasaki, Y. and Law, K. H. (1991). Generating qualitative models for structural engineering analysis and design. *Proceedings of the Seventh National Conference in Civil Engineering.*

Genet, R. M., Genet, D. R., Talent, D. L., Drummond, M., Hine, B., Boyd, L. J. and Trueblood, M. (1991). *Multi-use lunar telescopes.* Alexei V. Filippenko, (Ed.), Robotic Observatories in the 1990'. The Astronomical Society of the Pacific Conference Series.

Golding, A. and Rosenbloom, P. S. (1991). Improving rule-based systems through case-based reasoning. *Proceedings of the Ninth National Conference on Artificial Intelligence of AAAI,* Anaheim, CA:MIT Press.

Gruber, T. R. (1991). The role of common ontology in achieving sharable, reusable knowledge bases. *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, pp. 601-602.* Cambridge, MA: Morgan Kaufmann.

Gruber, T. R. and Russell, D. M. (1992). Computer-assisted formulation of engineering models: issues for knowledge acquisition. Stanford University, Knowledge Systems Laboratory. *Submitted to special issue on Modelling of International Journal of Intelligent Systems.*

Gruber, T. R. and Russell, D. M. (1992). *Derivation and use of design rationale information as expressed by designers.* In T. Moran & J.H. Carroll (Eds.), University of Michigan Department of Electrical Engineering and Computer Science.

Hanson, R., Stutz, J. and Cheeseman, P. (1991). Bayesian classification with correlation and inheritance. *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence. (IJCAI).* Sydney, Australia.

Iba, W. F. and Langley, P. (1992). Induction of one-level decision trees. *Submitted to the Tenth National Conference on Artificial Intelligence.*

Iwasaki, Y. (1991). *Towards an integrated model-based reasoning environment computer.* Kaguku, Vol. 1, No. 2, in Japanese.

Iwasaki, Y. (1992). Qualitative physics beyond automated mathematics. *To Appear in Computational Intelligence.*

Iwasaki, Y. and Simon, H. A. (1992). Retrospective on 'causality' in device behavior. *To appear in the Journal of Artificial Intelligence.*

Jang, J. S. (1991). ANFIS: Adaptive-network-based fuzzy inference systems. *Submitted to ITSMC.*

Kaelbling, L. P. (1992). *Learning in embedded systems,* MIT Press. Cambridge, Massachusetts (in press).

Kedar, S. and Chrisman, L. (1992). Mutual theory refinement: A case study. *Submitted to the Tenth National Conference on Artificial Intelligence.* San Jose, CA.

Knoblock, C., Minton, S. and Etzioni, O. (1991). Integrating abstraction and explanation-based learning in prodigy. *Proceedings of the National Conference on Artificial Intelligence.*

Laird, P. (1991-92). Discrete sequence prediction and its application. *In Computational Learning and Natural Learning Workshop. Submitted to the Tenth National Conference on Artificial Intelligence.*

Laird, P. (1992) Dynamic optimization. *Submitted to Machine Learning International Conference.*

Laird, J. E., Yager, E. S., Hucka, M. and Tuck, C.M. (1991). *Robo-soar: An integration of external interaction planning and learning using soar,* Robotics and Autonomous Systems. Elsevier North Halland, Vol. 8, 113-129.

Langari, R. Berenji, H. R. (1992). *Fuzzy logic in control engineering.* D. White and D. Sosege (Eds.), Handbook of Intelligent Control. Publisher: Van Nostrand.

Langley, P. (1992). Systematic and nonsystematic search strategies. *Submitted to the Tenth National Conference on Artificial Intelligence.*

Langley, P. and Allen, J. A. (1991). Learning, memory, and search in planning. *Proceedings of the Thirteenth Conference of the Cognitive Science Society,* pp. 364-369. Chicago: Lawrence Erlbaum.

Langley, P., Iba, W. F. and Thompson, K. (1992). An analysis of bayesian classifiers. *Submitted to the Tenth National Conference on Artificial Intelligence*

Lansky, A. L. (1992). Localization vs. abstraction: A comparison of two search reduction techniques. *Submitted to the Tenth National Conference on Artificial Intelligence.*

Lansky, A. L. (1991). Localized search for multiagent domains. *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI) pp. 252-258.* Sydney, Australia.

Levy, A. and Iwasaki, Y. (1992). Relevance reasoning to guide compositional modeling. *Submitted to the Tenth National Conference on Artificial Intelligence.*

Minneman, S. L. (1991). Managing a trois: A study of a multi-user drawing tool in design work. Presented at the Conference on Human Factors in Computing Systems. New Orleans, LA.

Minton. S., Bresina J. and Drummond, M. (1991). Commitment strategies in planning: A comparative analysis. *Proceedings of the Twelfth International Conference on Artificial Intelligence.*

Mooney, R.J. (1992). Induction over the unexplained: Using overly general domain theories to aid concept learning. *Machine Learning (in press).*

Mooney, R.J. (1991). *Explanation-based learning as concept formation in concept formation, knowledge and experience in unsupervised learning.* D. Fisher, M. Pazzani, and P. Langley (Eds.), San Mateo, CA: Morgan Kaufmann.

Muscettola, N., Smith, S. F., Cesta, A. and D'Aloisi, D. (1992). Coordinating space telescope operations within an integrated planning and scheduling architecture. *To appear in IEEE Control Systems Magazine.*

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T. and Swartout, W. R. (1991). Enabling technology for knowledge sharing. *AI Magazine, 12(3): 16-36.*

Newell, A., Yost, G. R., Laird, J. E., Rosenbloom, P. S. and Altmann, E. (1991). *Formulating the problem space computational model.* R. F. Rashid (Ed.), Science: A 25 year Commemorative. Addison-Wesley Press.

Ng, H. T. and Mooney, R. J. (1991). An efficient first-order horn-clause abduction system based on the ATMS. *Proceedings of the Ninth National Conference on Artificial Intelligence,* pp. 494-499. Anaheim, CA.

Nordhausen, B. and Langley, P. (1992). An integrated framework for empirical discovery. *Machine Learning. (In Press).*

Rosenbloom, P.S., Laird, J. E., Newell, A. and McCarl, R. A preliminary analysis of the foundations of soar. *Artificial Intelligence, Vol. 47,* pp. 289-325.

Rosenbloom, P.S., Newell, A. and Laird, J. E. (1991). *Towards the knowledge level in soar: The role of the architecture in the use of knowledge.* In K. VanLehn (Ed.), Architectures for Intelligence. Hillsdale, NJ: Lawrence Erlbaum.

Shavlik, J.W. , Mooney, R.J. and Towell, G. (1991). Symbolic and neural learning algorithms: An experimental comparison. *Machine Learning, 6,* 111-143.

Smith, S. F. and Pathak, D. K. (1992). Balancing anogonistic time and resource utilization constraints in over-subscribed scheduling problems. *Proceedings of the Eighth IEEE Conference on Artificial Intelligence Applications.*

Sun, C. T. et. al. (1991). Dynamic compensatory pattern matching in a fuzzy rule-based control system. *Proceedings of the American Control Conference.*

Sun, C. T., et. al. (1991). Protective device coordination expert system. *In the IEEE Transactions on Power Delivery. 6, 1,* 359-365.

Tang, J. and Leifer, L. (1991). An observational methodology for studying group design activity. *Journal of Research in Engineering Design, Vol. 2.*

Thompson, K. and Langley, P '1991). *Concept formation in structured domains.* In D. H. Fisher, M. J. Pazzani and P. Langley (Eds.), Concept Formation: Knowledge and Experience in Unsupervised Learning. San Mateo, CA: Morgan Kaufmann.

Wiesmeyer, M. D. (1991). An operator-based model of rapid visual counting. *In the Thirteenth Annual Conference of the Cognitive Science Society.* Chicago, IL.

Yager, R. R. and Zadeh, L. A. (Eds.) (1991). *Uncertainty in knowledge bases.* Bouchon-Meunier. Berlin:Springer-Berlag.

Zadeh, L. A. and Kacprzyk (Eds.)(1991). *Fuzzy logic for the management of uncertainty.* New York, NY: John Wiley and Sons.

Zadeh, L. A. (1992). Fuzzy logic and the calculus of fuzzy if-then rules. *To appear in AI Expert.*

Zweben, M., Davis, E. , Daun, B. and Deale, M. (1991). Scheduling and rescheduling with Iterative Repair. *To appear in the IEEE Systems, Man, and Cybernetics.*

Zweben, M., Davis, E., Daun, B., Drascher, E., Deale, M. and Eskey, M. (1991). Learning to improve constraint-based scheduling. *To appear in the Artificial Intelligence Journal.*

CONFERENCE/WORKSHOP PROCEEDINGS/TECHNICAL REPORTS

Allen, J. A. and Thompson, K. (1991). Probabilistic concept formation in relational domains. *Proceedings of the Eighth International Workshop on Machine Learning*, pp.375-379. Evanstan, IL: Morgan Kaufmann.

Baudin, C., Gevins J. (1991). A CA approach to design knowledge capture. *Proceedings of the AAAI Knowledge Acquisition Workshop*. Anaheim, CA.

Baudin, C. and Gevins, J. (1991). The influence of use on design rationale acquisition. *Proceedings of the Machine Learning Workshop*. San Mateo, CA: Morgan Kaufmann.

Baudin, C. and Gevins, J. (1991). A design rationale capture as knowledge acquisition. *Proceedings of the Machine Learning Workshop*. San Mateo, CA: Morgan Kaufmann.

Baudin, C., Gevins, J., Mabogunje, A. and Baya, V. (1991). A knowledge-based interface for design information retrieval. *In the Ninth Conference on Artificial Intelligence AAAI-91*, I Workshop on Intelligent Multi-Media Interfaces. Anaheim, CA.

Berenji, H. R. (1992). *Fuzzy inference with reinforcement learning*. Technical Report FIA-91-02. AI Research Branch, NASA Ames Research Center.

Berenji, H. R. (1992). Strategy learning in fuzzy logic control. *North American Fuzzy Information Processing Society Workshop*. Columbia, MO.

Berenji, H. R. (1991). Refinement of approximate reasoning-based controllers by reinforcement learning. *Proceedings of the Eighth International Workshop on Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Berenji, H. R. , Jani, Y. and Lea, R. (1991). Approximate reasoning-based learning and control for proximity operations and docking in space. *AIAA Guidance, Navigation and Control Conference*. New Orleans, LA.

Boy, G. A. (1991). Indexing hypertext documents in context. *Proceedings of Hypertext'91*. San Antonio, TX.

Boy, G. A. (1991). Some theoretical issues on the computer integrated documentation project. *Proceedings of the Sixth AAAI-Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop*. Banff, Canada.

Boy, G. A. (1991). *Computer integrated documentation*. Technical Memorandum 103870, NASA Ames Research Center.

Boy, G. A. (1991). On-line user model acquisition in hypertext documentation. *IJCAI Workshop on Agent Modeling of Intelligent interaction*. Sydney, Australia.

Boy, G. A. (1991). Use and acquisition of contextual knowledge in information retrieval. *AAAI Workshop on Knowledge Acquisition from Research to Application to Tools*. Anaheim, CA.

Boy, G. A. (1991). Integrated human-machine intelligence in space systems. *In the the Ninth IAA Man in Space Symposium*. Cologne, Germany.

Buntine, W.L. and Caruana., R. (1991). *Introduction to IND and recursive partitioning*. Technical Report FIA-91-28. AI Research Branch, NASA Ames Research Center.

Buntine, W.L. and Del Alto, M. (1991). *Collected notes on the workshop for pattern discovery in large databases (NASA Ames, January 14-15, 1991)*. Technical Report FIA-91-07. AI Research Branch, NASA Ames Research Center.

Carbonell, J. G, Etzioni, O., Gil, Y., Joseph, R., Knoblock, C. A., Minton, S. and Veloso, M. (1991). *An integrated architecture for planning and learning.* SIGART.

Carryer, J. E., Roy, R. H. and Powell, J. D. (1991). *Estimating in cylinder mixture temperatures using acoustic resonance.* Presented at the Symposium on Advanced Automotive Technologies ASME Winter Annual Meeting. Atlanta, GA.

Chapman, D. (1991). *Action and combinatorics.* Working Papers of the AAAI Fall Symposium on Sensory Aspects of Robotic Intelligence, Asilomar.

Cerbone, G. (1992). Machine learning techniques in optimal design. *Proceedings of the Second International Conference on Artificial Intelligence and Design.* Carnegie Mellon University. Pittsburgh, PA.

Cerbone, G. (1992). Inductive learning in engineering: A case study. *Proceedings of the Adaptive and Learning Systems. Conference of the IEEE Society for Optical Engineers on Artificial Intelligence (AAAI),* pp. 547-552. Anaheim, CA

Cerbone, G. and Dietterich, T. G. (1991). Knowledge compilation to speed up numerical optimization tasks. *Proceedings of the Eighth International Workshop on Machine Learning (ML-91),* pp. 600-604. Evanston, IL: Morgan Kaufmann.

Chapman, D. (1991). *On choosing domains for agents.* In collected notes from the benchmarks and metrics workshop (NASA Ames, July 25, 1990), Mark E. Drummond, Leslie P. Kaelbling and Stanley J. Rosenschein, (Eds.). Technical Report TR FIA-91-06. AI Research Branch, NASA Ames Research Center.

Choi, W. and Latombe, J. C. (1991). Planning and executing robot motions with incomplete knowledge. *IEEE/RSJ International Workshop on Intelligent Robots and Systems.* Japan.

Christiansen, A. D., (1992). *Automatic acquisition of task theories for robotic manipulation.* Doctoral Thesis. School of Computer Science, Carnegie Mellon University

Compton, M, Colombano, S. and Frainier, R. (1991). The astronaut science advisor: ground testing during SLS-1. *Proceedings of the Space Station Evolution Symposium.* (NASA Conference Publication 10083).

Drummond, M. (1991). An insider's guide to AI planning. Paper written to accompany a tutorial given at AAAI-91. Anaheim, CA.

Drummond, M., Bresina, J., Kedar, S. (1991). *The entropy reduction engine: Integrating planning, scheduling, and control.* SIGART

Drummond, M., Bresina, J., Swanson, K., Philips, A. and Levinson, R. (1991). *The APT/ERE planning and scheduling manifesto.* Technical Report TR-FIA-91-04. AI Research Branch, NASA Ames Research Center.

Fikes, R., Cutkosky, M., Gruber, T. Baalen, J. V. (1991). *Knowledge sharing technology project overview.* Technical Report KSL-91-71. Stanford University, Knowledge Systems Laboratory.

Fikes, R., Gruber, T., Iwasaki, Y., Levy, A. and Nayak, P. (1991). *How things work project overview.* Technical Report KSL-91-70. Stanford University, Knowledge Systems Laboratory.

Fruchter, R. Law, K. H. and Iwasaki, Y. (1991). An approach for qualitative structural analysis in artificial intelligence and structural engineering. *Proceedings of the Second International Conference on the Civil-Comp Press.*

Golding, A. (1991). *Pronouncing names by a combination of rule-based and case-based reasoning.* Doctoral Dissertation, Stanford University.

Gruber, T. R. (1991). *Ontolingua: A mechanism to support portable ontologies.* Technical Report KSL-91-17. Stanford University, Knowledge Systems Laboratory.

Gruber, T. R., Boose, J., Baudin, C. and Weber, J. (1991). Design rationale capture as knowledge acquisition: Trade offs in the design of interactive tools. *Proceedings of the Eighth International Workshop of Machine Learning,* pp. 3-12. Chicago, IL: Morgan Kaufmann.

Gruber, T., Tenebaum, J. M. and Weber J. (1991). *Toward a comprehensive representation for collaborative product development.* Technical Report KSL-91-78. Stanford University, Knowledge Systems Laboratory. *Submitted to AI and Design '92.*

Hammel, J. and Van der Loos, H. F. M. (1991). Clinical evaluation of robotics technology for use by persons with severe physical disabilities. *Proceedings ICORR'91 (International Conference on Rehabilitation Robotics).* Atlanta, GA.

Hammel, J. and Van der Loos, H. F. M. (1991). Factors in the prescription and cost-effectiveness of robot systems for high-level quadriplegics. *Proceedings of the 14th Annual RESNA (Rehabilitation Engineering Society of North America).* Kansas City, MO

Hanson, R., Stutz, J. and Cheeseman, P. (1991). *Bayesian classification theory.* Technical Report TR-FIA-90-12-7-01. AI Research Branch, NASA Ames Research Center.

Huffman, S.and Laird, J. E. (1991). *Perception, projection and reaction in external environments in soar: initial approaches.* Technical Report CSE-TR-118-91. Department of Electrical Engineering and Computer Science. University of Michigan.

Huffman, S., Pearson, D. J. and Laird, J. E. (1991). *Correcting imperfect domain theories: A knowledge-level analysis, in machine learning: Induction, analogy and discovery.* Technical Report CSE-TR-114-91. Chipman, S. and Meyrowitz, A. (Eds.), Luwer Academic Press.

Iba, W. F. (1991). *Acquisition and improvement of human motor skills: Learning through observation and practice.* Doctoral Dissertation, Department of Information and Computer Science, University of California, Irvine.

Iba, W. (1991). Modeling the acquisition and improvement of motor skills. *Proceedings of the Eighth International Workshop on Machine Learning, pp. 60-64.* Evanstan, IL: Morgan Kaufmann.

Iwasaki, Y. and Low, C. M. (1991). *Model generation and simulation of device behavior with continuous and discrete changes: The details.* Technical Report KSL-91-30. Stanford University, Knowledge Systems Laboratory.

Jang, J. S. (1991). A self-learning fuzzy controller with application to automobile tracking problem. *Proceedings of the IEEE Roundtable Discussion on Fuzzy and Neural Systems, and CIE Application.* Institute of Industrial Science, University of Tokyo, Japan.

Jang, J. S. (1992). Fuzzy controller design with domain experts. *To appear in the Proceedings of IEEE International Conference on Fuzzy Systems.*

Jang, J. S. (1991). Fuzzy modeling using generalized neural networks and {K}alman filter algorithm. *Proceedings of the AAAI-91 Workshop on Automating Software Design.* Anaheim, CA.

Jang, J. S. (1991). Rule extraction using generalized neural networks. *Proceedings of the Fourth IFSA World Congress.*

Kanefsky, B., Stutz, J. and Cheeseman, P. (1991). *An automatic classification of a landsat/TM image from Kansas (FIFE)*. Technical Report TR-FIA-91-26. AI Research Branch, NASA Ames Research Center.

Kedar, S., Bresina, J. and Dent, L. (1991). The blind leading the blind: Mutual refinement of approximate theories. *In the Proceedings of the Eighth International Workshop on Machine Learning*. Evanston, IL.

Kedar, S. and McKusick, K. (1992). *There is no free lunch: Tradeoffs in the utility of learned knowledge*. Technical Report TR-FIA-92-04. AI Research Branch, NASA Ames Research Center.

Keller, Richard M. (1991). Building the scientific modeling assistant: An interactive environment for specialized software design. *Proceedings of the AAAI-91 Workshop on Automating Software Design*. Anaheim, CA.

Khedkar, P. and Jhingran, A. (1992). Analysis of recovery in a database system using a write-ahead log protocol, IBM Technical Report. *Submitted to SIGMOND*.

Khedkar, P. and Keshav, S. (1991). Fuzzy prediction of timeseries. *To appear in the Proceedings of Fuzzy IEEE*. San Diego, CA.

Kulkarni, D. and Kay, B. (1992). Diagnosis of shuttle reaction control system. *In the Proceedings of NASA Monitoring and Diagnosis Workshop*.

Kulkarni, D. and Robinson, P. (1992). An expert system for monitoring dynamic aspects of RCS. *To Appear in AIAA-92 Proceedings*.

Laird, J.E., Hucka, M., Huffman, S. B., and Rosenbloom, P. S. (1991). *An analysis of Soar as an integrated architecture*. SIGART Bulletin, 2, 98-103.

Laird, J. (1991. *Preface to the Proceedings of the AAAI Spring Symposium on Integrated Intelligent Architectures*. Palo Alto, CA.

Laird, P. (1992). *TDAG: an algorithm for learning to predict discrete sequences*. Technical Report FIA-92-01. AI Research Branch, NASA Ames Research Center.

Laird, P. (1991). *Explanation-based generalization: Theory meets experiment*. Technical Report FIA-91-01-10-7. (Text of an invited lecture delivered to JSAI Symposium, Tokyo, Japan.)

Langley, P. (1991). Machine learning and language acquisition. *Proceedings of the AAAI Spring Symposium on Machine Learning of Natural Language and Ontology*. Stanford, CA.

Langley, P. and Allen, J. A. (1991). The acquisition of human planning expertise. *Proceedings of the Eighth International Workshop on Machine Learning*, pp. 80-84. Evanston, IL: Morgan Kaufmann.

Langley, P., McKusick, K. B., Allen, J. A., Iba, W. F. and Thompson, K. (1991). A design for the Icarus architecture. *Proceedings of the AAAI Spring Symposium on Integrated Intelligent Architectures*. Stanford, CA.

Leifer, L. (1991). Development of a visual language for assisted robot command-control. *Proceedings of the Cambridge Rehabilitation Robotics Workshop*.

Leifer, L. (1991). Instrumenting the design process: For real-time text-graphic design process records. *Proceedings of the International Conference on Engineering Design*, pp. 314-321.

Leifer, L., Van der Loos, H. F. M. and Lees, D. (1991). Visual language programming: For robot command-control in unstructured environments (plenary lecture). *Proceedings of the Fifth International Conference on Advanced Robotics*, pp. 31-36. Pisa, Italy.

Lowry, M. and Kedar, S. (1991). Toward a learning apprentice for software re-engineering. *In the Proceedings of the Ninth National Conference on Artificial Intelligence, Workshop on Automating Software Design.* Anaheim, CA.

Mahoney, J.J. and Mooney R.J.(1991). *Initializing ID5 with a domain theory: Some negative results.* Technical Report AI91-154, Artificial Intelligence Laboratory, University of Texas at Austin.

Minneman, S. L., and Bly, S. A. (1991). Managing a Trois: A study of multi-user drawing tool in distributed design work. *Proceedings of the Conference on Human Factors in Computing Systems,* pp. 217-224. Reading, MA: Addison-Wesley.

Minneman, S. L., and Tang, J. C. (1991). Video whiteboard: Video shadows to support remote collaboration. *Proceedings of the Conference on Human Factors in Computing Systems,* pp. 315-322. Reading, MA: Addison-Wesley.

Minneman, S. L. and Tang, J. C. (1991). *Video whiteboard: Video shadows to support remote collaboration.* Presented at the Conference on Human Factors in Computing Systems. New Orleans, LA.

Minneman, S. L. (1991). *The social construction of a technical reality: empirical studies of group engineering design practice.* Stanford Dissertation Defense. Stanford, CA.

Minton, S. (1991). *On modularity in integrated architectures.* SIGART

Minton, S., Bresina, J. and Drummond, M. (1991). Commitment strategies in planning: A comparative analysis. *Proceedings of the International Joint Conference Artificial Intelligence (IJCAI).* Sydney, Australia.

Minton, S., Bresina, J., Drummond, M. E. and Philips, A. B. (1991). *An analysis of commitment strategies in planning: The details.* Technical Report TR-FIA-91-08. AI Research Branch, NASA Ames Research Center.

Minton, S., Knoblock, C. A., Etzioni, O., Carbonell, J. G. and Veloso, M. (1991). Evaluating the integration of EBL and abstraction in PRODIGY. *Workshop on Empirical Machine Learning,* Keio University, Japan.

Mahoney, J. J. and Mooney, R. J. (1991). *Initializing ID5 with a domain theory: Some negative results.* Technical Report AI91-154. Artificial Intelligence Laboratory, University of Texas at Austin.

Mooney, R.J. and Ourston, D. (1991). *Theory refinement with noisy data.* Technical Report AI91-153. Artificial Intelligence Laboratory, University of Texas at Austin.

Mooney, R.J. and Ourston, D.(1991). Constructive induction in theory refinement. *Proceedings of the Eighth International Machine Learning Workshop,* 178-182. Evanston, IL.

Mooney, R.J. and Ourston, D. (1991). A multi-strategy approach to theory revision. *Proceedings of the International Workshop on Multi strategy Learning,* 115-130. Harpers Ferry, West Virginia.

Muscettola, N. (1992). *Scheduling by iterative partitioning of bottleneck conflicts.* Technical Report. The Robotics Institute, Carnegie Mellon University.

Muscettola, N. and Smith, S. F. (1992). Decomposability and scalability in space-based observatory scheduling. *Proceedings in the AAAI Spring Symposium on Practical Approaches to Scheduling and Planning.*

Nayak, P., Joskowicz, L. and Addanki, S. (1991). *Automated model selection using context-dependent behaviors.* Technical Report KSL-91-16. Stanford University, Knowledge Systems Laboratory.

Nishihara, K. H. and Chapman, D. (Contributors) (1991). Promising directions in active vision. Michael J. Swain and Markus Stricker (Eds.). Technical Report CS-91-27. University of Chicago. Chicago, IL.

Oakley, C. M. (1991). *Experiments in modelling and end-point control of two-link flexible manipulators.* Stanford University Ph.D. Thesis. Stanford, CA.

Ourston, D. and Mooney, R.J. (1991). Improving shared rules in multiple category domain theories. *Proceedings of the Eighth International Machine Learning Workshop,* pp. 534-538. Evanston, IL.

Ourston D. (1991). *Using explanation-based and empirical methods in theory revision.* Technical Report AI91-164. Artificial Intelligence Laboratory, University of Texas at Austin. Doctoral Dissertation.

Ourston, D. and Mooney, R. J. (1991). Improving shared rules in multiple category domain theories. *Proceedings of the Eighth International Machine Learning Workshop,* pp.534-538. Evanston, IL.

Philips, A. and Bresina, J. (1991). *NASA TileWorld manual.* Technical Report TR-FIA-91-11. AI Research Branch, NASA Ames Research Center.

Philips, A., Swanson, K., Drummond, M. and Bresina, J. (1991). *A design rationale for NASA TileWorld.* Technical Report TR-FIA-91-04. AI Research Branch, NASA Ames Research Center.

Richards, B.L. and Mooney R.J. (1991). First-order theory revision. *Proceedings of the Eighth International Machine Learning Workshop,* pp. 447-451. Evanston, IL.

Subramanian, S. and Mooney, R.J. (1991). Combining abduction and theory revision. *Proceedings of the International Workshop on Multi strategy Learning,* pp. 207-214. Harpers Ferry, W. VA.

Sun, C. T. (1991). Interpolative reasoning in distributed intelligent systems based on fuzzy sets and neural nets. *Proceedings of the IFAC Symposium on Distributed Systems.*

Thompson, K., Langley, P. and Iba, W. F. (1991). Using background knowledge in concept formation. *Proceedings of the Eighth International Workshop on Machine Learning,* pp. 554-558. Evanston, IL: Morgan Kaufmann.

Utgoff, P. E. and Brodley, C. E. (1991). *Linear machine decision trees.* Technical Report 91-10. Department of Computer Science. University of Massachusetts. Amherst, MA.

Van der Loos, H. F. M., Hammel, J. and Leifer, L. (1991). Robot effectiveness: Enhancing the independence of quadriplegics in the workplace. *Proceedings of the Fifth International Conference on Advanced Robotics,* pp. 910-914. Pisa, Italy.

Whalen, T. and Berenji, H. R. (1991). Actions as evidence: Multiple epistemic agents acting under uncertainty. *Second IEEE Annual Conference on A I simulation and Planning in High Autonomy Systems.* Cocoa Beach, FL.

Wiesmeyer, M.D. (1992). *An operator-based model of covert visual attention.* Doctoral Dissertation. The University of Michigan, Ann Arbor.

Zadeh, L. A. (1991). Fuzzy logic and the calculus of fuzzy if-then rules. *In the Proceedings of SYNAPSE.* Tokyo, Japan.

Zweben, M., Davis, E., Daun, B. and Deale, M. (1992). Rescheduling with iterative repair. *Proceedings of the AAAI Spring Symposium on Practical Approaches to Scheduling and Planning.* Stanford, CA.

Zweben, M. and Davis, E. (1992). Learning to improve iterative repair scheduling. *To appear in Machine Learning 1992 Workshop on Knowledge Compilation and Speedup Learning.*