UPDATES

We welcome comments, and apologize for errors made on previous issues, of the SIGART Newsletter. The following errors/corrections and updates (SIGART #81 July 1982) were brought to our attention by Eunice Edwards, Publications Secretary, of the Imperial College of Science and Technology, University of London.

1. Publication 81/6 "Logic as a Computer Language of Children: A One Year Course" is no longer available as a report, but as a booklet; the price of which is 3 pounds. Publications 81/6 and 81/8 will now be included in Mr. Ennals' new book "Beginning Micro-Prolog" to be issued in November 1982. The publishers will be Heinemann Education in Computing and Ellis Horwood Ltd. and the price will approximately be 5 pounds (plus postage charges to the United States).
2. Publication 81/4 "An Introduction to Logic Programming" by Keith Clark should read 81/14.
3. The latest publications 82/3 "Logic Program Specification of Numerical Integration," by K.L. Clark, et al. and publication 82/4 "Logic Programming for Expert Systems," by Ph. Hammond cost .56 pounds and 3.50 pounds, respectively

SPECIAL SECTION ON PLANNING

RESEARCH ON PLANNING

Ann E. Robinson
SYMANTEC
306 Potrero
Sunnyvale, Calif

This section of the newsletter contains a collection of summaries of current research on planning. Although the number of projects described is modest, the variety of topics and institutions represented illustrates a healthy and growing interest in this area. In contrast with early research, much of which dealt with "blocks-world" problems, a significant portion of the research described here is confronting so-called "real-world" problems, such as planning with incomplete information, planning with large numbers of constraints, and planning in a non-cooperative environment. The role of planning in communication is also an important topic. Furthermore, in addition to covering a broad range of topics, the research covers the spectrum from quite theoretical to fairly practical.

The articles are arranged alphabetically by institution. They have been reproduced essentially as they were submitted, with some minor corrections and formatting changes. I apologize to the authors for any errors that may have been introduced in the process.

Since I know these summaries do not cover all the current planning research, I would like to encourage others doing related research to send a summary to SIGART or me to be published in subsequent issues.

I would like to thank Dan Russell for helping solicit articles and helping edit and proofread them.

Ann Robinson

Planning Research at the Intelligent Systems Laboratory

The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania

ISIS

Mark Fox, Brad Allen, Stephen Smith, Gary Strohm.

Introduction. ISIS is a constraint-directed planning/scheduling system for factory job-shops. Research began during the summer of 1980 in conjunction with the Westinghouse turbine component plant in Winston-Salem NC. Our research has resulted in two systems being constructed, ISIS-I and ISIS-II [1].

Application Domain. The plant under consideration presents one of the most complex types of scheduling tasks. The plant produces thousands of different part numbers, some of which are similar, others of which are not. Any part can be ordered in quantities from one to hundreds. Each part number has one or more process routings containing at least ten operations. A process routing may differ simply in machine substitutability, or may represent a totally different manufacturing process. Each operation in a process routing requires resources such as machines, tools, operators, fixtures, and materials. At any time there are over 200 orders in the plant, each competing for the same resources.

This scheduling problem has been described as NP-hard. The simple sequencing (without gaps or alternative routings) of 10 orders on 5 machines can result in $(10!)^5$ possible schedules. Rather than do simple capacity analysis, as found in the majority of scheduling systems, or a local dispatch rule approach, as found in operations management research, a constraint-directed reasoning approach was chosen. It was found that schedulers spend 80%-90%

of their time determining the constraints in the environment that will affect their scheduling decision, and 10%-20% of their time actually constructing and modifying schedules. Any system that is to adequately schedule such an environment must attend to the multitude and variety of constraints.

Issues. The construction of a system to plan and schedule an operation sequence for each order must face the following issues:

- □ How to represent the variety of constraints and their relaxations.
- □ How to satisfy (possibly conflicting) constraints.
- □ Reasoning about time in the scheduling of operations.
- □ Differential application of constraints.
- □ The size/complexity of the search space.

Research Description. ISIS-II's knowledge representation language, SRL, provides the capability to represent a variety of constraints and their alternative relaxations. Scheduling is performed by a multi-pass beam search. Constraints are used in a pre-search analysis to select search operators, hence bounding the size of the search space. Search operators generate alternative partial schedules by choosing operations, machines, shifts, materials, etc. Each alternative represents a relaxation of a constraint. Constraints are also used to rate alternative partial schedules; defining which states are in the beam. ISIS-II selects the constraints to rate a state, and differentially applies them as defined by scheduling goals. A rule-based post-search analysis determines if any of the alternative schedules satisfy the order's constraints. If not, pre-search is informed and a new search space is defined and searched.

Future Research. A third version, ISIS-III, is currently under investigation. It takes a Hearsay-II like approach to scheduling. Schedules will be constructed opportunistically. Separate knowledge sources will smooth resource utilization according to the existing constraints.

Resource-Limited Planning with Incomplete Information

Mark Fox

Introduction. Most planning research assumes domains with perfect information: operator and state descriptions completely describe the problem and all possible solutions, and planning algorithms search until a plan is found, without concern for the resources consumed (e.g., time or space). To date, little research has been performed in planning with incomplete information with the added constraint of limited resources. This research investigates issues of attention focusing and knowledge acquisition when the planning process is constrained by limited resources [1].

Application Domain. The problem of resource-limited planning/reasoning was investigated in the general domain of rule-based systems. Viewing each rule as a operator, satisfying its condition may be arbitrarily difficult, if not impossible. A production rule may require information that is not in the knowledge base to completely evaluate its condition. In order to evaluate its condition, further knowledge would have to be acquired (e.g., inferred or deduced).

Research Description. A production system was constructed in SRL, a knowledge representation language. A rule's condition could take the form of any SRL pattern or lisp function. In evaluating a rule's condition, there may not exist enough information in the knowledge base to evaluate it completely. Hence, the task was to select a rule and its conditions to investigate further, i.e., to acquire enough knowledge to completely evaluate it, under the constraint of limited resources.

The approach taken in this research was to construct schemata which described alternative methods and their resource costs for evaluating patterns in a rule's condition. The system took a least cost approach to selecting and evaluating rules. The use of a method to evaluate a condition pattern was annotated and used to guide the system in its decision to further investigate the current condition or another rule's condition.

References

1. Fox M.S., "Reasoning With Incomplete Knowledge in a Resource-Limited Environment: Integrating Reasoning and Knowledge Acquisition", *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver BC, Canada, Aug. 1981.
2. Fox M.S., B. Allen, and G. Strohm, "Job-Shop Scheduling: An Investigation of Constraint-Directed Reasoning", *Proceedings of the American Association for Artificial Intelligence*, Pittsburgh PA, August 1982.

Counterplanning and Planning by Analogy

Jaime G. Carbonell
Computer Science Department
Carnegie-Mellon University

For the past 6 years or so I have been analyzing planning strategies and methods that had not heretofore been addressed in AI work on planning. In particular, planning in dynamically-changing adversary situations required the formulation of new planning mechanisms. More recently, I have focused on the problem of recycling and transforming planning structures by reasoning analogically from past experience where plans were formulated and tested in similar situations.

Counterplanning in the POLITICS project

POLITICS is a system that simulates ideologically-based reasoning in the domain of international politics. An interesting finding stemming from the POLITICS work is that given a flexible system for planning against adversaries (i.e. counterplanning), it suffices to attribute different goal hierarchies to different political planners in order to produce radically divergent plans and expectations. The counterplanning system itself remained invariant across different simulated political ideologs.

Counterplanning in POLITICS comes in two flavors: 1) *Obstructive counterplanning* consists of planning to thwart the inferred objectives of an adversary by monitoring, inferring and interfering with his or her plans, and 2) *constructive counterplanning* is the dual process, that is, planning to achieve one's objectives in light of expected interference from potential adversaries. The counterplanning strategies are essentially (rather powerful) heuristic rules that exploit known goals and causal structures of plans as they guide the search for effective counterplans. Counterplanning in the context of the POLITICS system is discussed in some depth in [1], [2], and [3].

Formulating and Generalizing Plans by Analogy

People seldom plan or solve problems without appealing to past experience. Typically, there are general plans for recurring situations, but when these fail or do not exist, it is more typical and often more economical to retrieve plans formulated and tested in similar situations, and subsequently modify these plans to suit current needs. Moreover, one can, on occasion, construct useful analogies bridging radically different domains to formulate plans in what otherwise might prove to be knowledge-poor situations, untractable to more conventional planning techniques.

I have been developing a method of problem-solving and planning by analogy based on an extension of means-ends analysis and a (MOPS-like) model of reconstructive memory. Additionally, successfully analogized plans, together with causal analysis of unsuccessful analogies (that highlight crucial differences between the old and new planning scenarios) provide exactly the input necessary for a learning-from-examples module, enabling near-miss analysis and establishing invariant causal structures. The analogical planning work is very much in progress and results are encouraging but preliminary. In contrast, the counterplanning work reached a measure of completeness and has recently been relegated to the proverbial back burner. More details on the current state of the analogical reasoning work can be found in [4,], [5], and [6].

References

1. Carbonell, J. G., *Subjective Understanding: Computer Models of Belief Systems*, Ann Arbor, MI: UMI Research Press, 1981.
2. Carbonell, J. G., "Counterplanning: A Strategy-Based Model of Adversary Planning in Real-World Situations," *Artificial Intelligence*, Vol. 16, 1981 , pp. 295-329.
3. Carbonell, J. G., "POLITICS: An Experiment in Subjective Understanding and Integrated Reasoning," in *Inside Computer Understanding: Five Programs Plus Miniatures*, R. C. Schank and C. K. Riesbeck, eds., New Jersey: Erlbaum, 1981.
4. Carbonell, J. G., "Learning by Analogy: Formulating and Generalizing Plans from Past Experience," in *Machine Learning, An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell and T. M. Mitchell, eds., Tioga Press, Palo Alto, CA, 1982.
5. Carbonell, J. G., "A Computational Model of Problem Solving by Analogy," *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pp. 147-152, August 1981.
6. Carbonell, J. G., "Experiential Learning in Analogical Problem Solving," *Proceedings of the Second Conference of the American Association for Artificial Intelligence*, Pittsburgh, PA. 1982.

Artificial Intelligence System for Administration (AISA)

University of Costa Rica AI Group
Escuela de Ciencias de la Computacion e Informatica
Universidad de Costa Rica
Ciudad Universitaria Rodrigo Facio
Costa Rica

Claudio Gutierrez, Coordinator,
Eduardo Piza, Francisco Mata
Vladimir Lara, Michael Pengelli (Open University, UK)
Alvaro de la Ossa

AISA is an ambitious project of interactive, distributed planning of administrative acts. AISA will be developed in several stages. The envisioned final system will consist of an administrative network of personal computers, each one dedicated to serve a particular job. Each workstation in the network will be "manned" by a synergic person-computer pair which will be responsible for the specific job. Each half of the pair will take care of the functions best suited to his/her/its nature: the computer will take care of housekeeping and accounting, including electronic mail, electronic archive, regulation updating and truth maintenance; the human will be responsible for the actual decision making.

It is hoped that we can bypass the problem of natural-language interface by having the users communicate in logical clauses that represent the goals to be accomplished. The environment will be one of "distributed programming" in which each workstation will decompose one goal at a time (in Prolog-like fashion). Fulfillment of the goals could then be delegated to different workstations. This "shallow programming" will be compatible with the lack of computing expertise on the part of the users ("weak programmer"). The whole system could be conceived of as an expert system -- the organization -- made of several "weak experts" -- the workstations.

We are planning to write the long-range program for the system in GOAL, an AI programming language developed by Claudio Gutierrez based on Dan Chester's HCPRVR. GOAL, in a Spanish-speaking version, is already implemented on our B-6900 computer, which will simulate the workstations through time-sharing. Plans are underway for the production of low-priced personal computers by the School of Electrical Engineering. These personal computers will host the language as special-purpose machines. GOAL is a Prolog-like interpreter in LISP.

The first stage of the project, which we are now in, is limited to the implementation of electronic mail, electronic archive, interactive tracing of documents, and a moderately intelligent task-management facility. A running program, under the name PROTOASIA, with a modicum of the capabilities for the task-management facility, is already available. Since this is a new AI group, we hope that the experience we will gain with the satisfaction of the modest goals of the first stage will be instrumental in the acquisition of skills that we need for the major undertaking remaining.

The UCR AI group maintains close ties with the AI groups of the University of Delaware, Edinburgh University, the Open University (UK), and Essex Univeristy.

References

1. Gutierrez, C., Descripcion del proyecto AISA: Archivo Inteligente para Seguimiento Administrativo, UCR AI Memo 1. Junio 1980.
2. Gutierrez, C., An AI System for Administration, UCR AI Memo 2. January 1981.
3. Gutierrez, C., Presentation of AISA project to Dr. Jim Howe, (Edinburg University), UCR AI Memo 3. May 1981.
4. Gutierrez, C., Artificial Intelligence for a developing country. UCR AI Memo 4. November 1981.
5. Gutierrez, C., GOAL: manual del usuario, UCR AI Memo 5. Julio 1982.
6. Gutierrez, C., E. Piza, and F. Mata, El Proyecto AISA: diseno para la primera fase. UCR AI Memo 6. Agosto 1982.
7. Pengelli, M. AISA Pilot Systems, UCR AI Memo 7., August 1982.
8. Mata, F., Opiniones sobre "AISA Pilot Systems". UCR AI Memo 8. August 1982.

Planning Systems Research

M.I.T., Laboratory for Computer Science

Albert Vezza, J.C.R. Licklider,
Lowell Hawkinson, Stuart Galley,
P. David Lebling, Christopher Reeve

Our research in planning is directed at investigating the concepts and ideas required for a strategic planning system (PS). As a result of this work, we expect to develop a strategic planning system, as well as the requisite technology for its creation and maintenance in a distributed environment.

Initially, the PS will be used in the planning and management of resources allocated to research and development. However, the planning system, by virtue of its generality, is appropriate for the planning and management of a broad class of resources applied to any one of a correspondingly broad set of goals. The planning system will be particularly suitable for hierarchically-structured, geographically-distributed large team efforts, where there might be local autonomy of the controlled resources at the leaves of the hierarchy. The system will also be suitable for the collection and aggregation of information, relative to planning and resource management, from geographically distributed sources, and for the creation of deployment models from which a set of deployment directives can be generated.

A key objective of our research is to have a planning system that is easy to use. The user interface to the planning system will be based on a simple yet powerful semantic model: the use of tables or work-sheets to hold the objects and programs that a user needs for planning and communication. In fact, the planning system may be viewed by its users as an extension of the spread-sheet class of systems typified by VisiCalc and SuperCalc. PS will apply tabular semantics to (1) data base queries; (2) intercommunication among users and processes in a distributed environment; (3) arbitrarily complex programs; and (4) management of mixed media. *Programs* may be active parts of tables, in that they can cause arbitrary computations to be performed on the entries of one or more tables. PS will support communication among table owners by *messages*, which are particular types of tables, and by links, which couple parts of tables.

Planning aids (PLAIDs) are subsystems of PS which provide specialized services to planners. PLAIDs may be (1) *input/output oriented,* (2) *computationally oriented,* or (3) *knowledge base oriented.* A knowledge-based PLAID might take a partially completed hypothetical plan and complete it using inference and previously completed plans, or it might take a high-level specification for a PLAID and generate a PS program that implements that specification.

Accomplishments to date include (1) the design and partial implementation of the planning system and (2) the development of knowledge representation conventions and reasoning capabilities for the knowledge-based PLAIDs. A knowledge representation system called PREP has been developed. It possesses reasoning capabilities relating to inheritance, maintenance of viewpoints, decomposition, plan evaluation, and time. PREP has been used to develop a knowledge-based PLAID that encourages the use of structure in the planning process.

References

1. Hawkinson, Lowell B., The PREP Knowledge Representation System. Technical Memo., M.I.T. Laboratory for Computer Science. (Forthcoming)
2. Michalek, Thomas F., A Rigorous Approach to Some Basic Inference Problems. Master's Thesis, M.I.T. June, 1982.
3. Ross, Steven I., A Plan Evaluation System. Master's Thesis, M.I.T. September, 1982.

Plans and the Real World

Arthur M. Farley
Department of Computer and Information Science
University of Oregon
Eugene, Oregon 97403

The main goals of the research program are to characterize real-world contexts and to investigate the implications such contexts have for the processes of plan generation and plan execution.

Real-world contexts can be characterized by properties that include the following: the context is ongoing before a goal arises and continues after it is satisfied; an agent has only imprecise (incomplete and inaccurate) understanding of context situations; there exist interacting agents and other systems that can affect context situations; there are limitations on physical, energy, and time resources.

Although the above properties offer no surprises, they are exactly the opposite of assumptions made by traditional problem-solving research investigating puzzle-like contexts. The properties of real-world contexts indicate that plans need to be robust and incompletely specified. Plan execution must involve opportunistic selection and tactical completion (or correction) of possible next-steps of pending plans. Plan execution depends upon the interpretation of results of continual information gathering activity. Furthermore, measures which improve the likelihood of favorable situations are suggested. Such measures include: controlling (e.g., owning) an area where plan execution is to occur; maintaining an area in a standard configuration (i.e., cleaning up); prior coordination of activities involving multiple agents (i.e., scheduling).

Accomplishments.

Most of our research has been theoretical in nature, leading to several publications. In [1], we discuss an opportunistic method for coordinating the satisfaction of multiple goals by merging the execution of previously known, general solution plans. In [2], waiting is defined as the activity whereby a subset of preconditions for an action are maintained in expectation that others will be satisfied without direct intervention. We describe situations that suggest waiting and discuss information relevant to deciding whether to wait.

In [3,4], we describe a probabilistic generalization of problem space and investigate its application to problem solving under conditions of uncertainty. An uncertain state is modeled as a set of state descriptions with associated probabilities summing to 1.0. Unreliable operators are probabilistic Markov processes. A technique for incremental planning, interspersing plan execution with plan generation, is defined, employing a notion of expected degree of goal state satisfaction associated with a plan tree.

Current Focus and Future Goals.

At present, time is the primary focus of our research effort. Representing and reasoning about the temporal aspects of events, activities, and plans will be a necessary component of real-world planning systems. This fact has generated much interest in time-related issues, as exemplified by the recent work of P. Hayes, D. McDermott, S. Vere, and J. Allen, among others.

Our approach is to represent the notion that at any point in time there are a set of CONSTANTs and a set of CHANGINGs that hold. Every CONSTANT and CHANGING has an associated propositional content, as well as a beginning and ending EVENT. Each EVENT has a time, represented as an INSTANT. An INSTANT embodies constraints on the time after which its EVENT must occur and on the time before which it must occur. Current upper and lower bounds on these values are maintained. Constraints are placed by making assertions about temporal relationships among the beginning and ending events of CONSTANTs and CHANGINGs. A PROCESS (PLAN) is represented as sets of CONSTANTs, CHANGINGs, and temporal relationships among them.

Work is currently proceeding on implementing the basic machinery needed, such as constraint propagation and consistency checking. The long-term goal will be to develop an interactive planner and execution monitor for scheduled plans.

References

1. Farley, A.M., "Issues in knowledge-based problem solving", I.E.E.E. Transactions on Systems, Man, and Cybernetics, August, 1980.
2. Farley, A.M., "On waiting", Proceedings of the First National Conference on Artificial Intelligence, Stanford, CA, August, 1980.
3. Farley, A.M., "A probabilistic model for uncertain problem solving", AI Center Tech. Note 256, SRI Inrnational, Menlo Park, CA; to appear in I.EE.E. Transactions on System, Man and Cybernetics, 1983.
4. Farley, A.M., "Incremental Planning in a probabilistic model for uncertain problem solving", Proceedings of the Fourth Conference of the Canadian Society for the Computational Study of Intelligence, Saskatoon, Sask., May, 1982.

Work on Planning and Problem Solving at Rochester

James Allen, Patrick Hayes,
Henry Kautz, Hans Koomen,
Andy Haas (now at BBN)

Computer Science Department
University of Rochester
Rochester, NY 14627

The work on planning at Rochester is mostly concerned with world models and world reasoning to support planning, rather than on finding new techniques for generating plans.

Many of the issues we are investigating are motivated by ARGOT, the Rochester dialogue system. In particular, in that project we need to construct, recognize, and generally manipulate plans that involve social interaction between multiple agents, the most important interaction being verbal communication.

The plans that we manipulate contain many actions other than purely physical actions. In particular, we must deal with communicative actions (speech acts) and mental actions such as inference. Thus one might plan to acquire information and then use that information to infer some new knowledge. One might plan to get another agent to make certain inferences. Recent work by Haas that addresses these issues using a syntactic model of beliefs is found in the references. Some of the dialogues that we are studying actually discuss plans for accomplishing tasks. For example, people request others to perform a plan and to modify existing plans. To support this we need to be able to refer to plans as objects, and to be able to reason about plans with incomplete knowledge. This project is still in its infancy, but we can outline our progress so far.

Viewing plans as objects to be reasoned about and discussed raises many issues. The most important of these is determining what plans consist of and what properties they have. Our initial answer to this is quite vague and needs considerable elaboration. A plan is simply a collection of actions, events, and states that are causally related. One view is that a plan summarizes a hypothetical simulation of some part of the world. To support this view, we need to identify a collection of concepts and a supporting representation in which we can specify and reason about quite general actions and events. This requires a representation that can handle temporal and spatial reasoning. Progress in these areas is reported in the references.

Our immediate goals include the following:

- Building a problem solver using our temporal representation. Initially, we intend to cover problems that can be solved by existing systems, and then to investigate worlds (blocks worlds) in which objects may move in more or less predictable ways (e.g., on conveyor belts, or spinning tables).
- Investigating multi-agent planning models in which the agents are not assumed to be cooperating. They may be competing or simply pursuing different goals that happen to require interaction. Such models are essential to analyzing dialogues in which such non-cooperative behavior occurs.

References

1. Allen, J.F., "A general model of action and time," TR 97, Computer Science Dept., U. Rochester, November 1981.
2. Allen, J.F., "An interval-based representation of temporal knowledge," Proc., 7th Int'l. Joint Conf. on Artificial Intelligence, Vancouver, B.C., August 1981.
3. Haas, A., "Planning mental actions," TR 107, Computer Science Dept., U. Rochester, July 1982.
4. Hayes, P.J., "Naive physics I: Ontology for liquids," Working Paper 63, Institut pour les Etudes Semantiques et Cognitives, Geneva, 1978.
5. Hayes, P.J., "Histories: A representation for time and change," to appear in J.R. Hobbs (Ed). Contributions in Artificial Intelligence, Vol. 1. Norwood, NJ: Ablex Publishing Corporation, 1982.

SHEM -- A Schema-Based Problem Solver

Daniel Russell
University of Rochester,
Rochester, New York
(currently at Xerox PARC)

I'm building a system, named SHEM, to construct assemblies of Fischer-Technik blocks. Goals are expressed as solid objects to be approximated by Fischer-Technik constructions. Fischer-Technik blocks have a few properties that set this domain apart from the classical "simple" blocks world. Instead of planar faces, blocks have projections or depressions that can be matched to form relatively rigid attachment sites. This places many additional constraints on the ways in which assemblies can be made, and demands more work from the planner in order to avoid conflicting projections and unmatched blocks.

SHEM explores the idea of schema-based problem solving; in essence, using pre-existing abstract plan fragments to construct a problem solution. In so doing, SHEM uses hierarchical and constraint-satisfaction problem-solving techniques rather than explicit operator search. SHEM also incorporates an explicit representation of actions and events over time. Unlike other planning systems, this representation allows strong distinctions to be drawn between past actions, planned actions, and causality.

In addition, SHEM is being built to study schema-derived solutions to problems of limited resources (constraints on time and block type) and problem solving in a dynamic and contrary world.

SHEM is being implemented in INTERLISP-D on the Xerox D machines. SHEM is currently about half-built, and is expected to be completed by the end of January, 1983.

References

This work will be available as a University of Rochester Technical Report in the Spring of 1983.

Research on Planning at SRI International

David Wilkins, Editor
David Wilkins, Doug Appelt,
Kurt Konolige, Richard Waldinger,
Stan Rosenschein, Mabry Tyson,
Tom Garvey

Artificial Intelligence Center
SRI International
Menlo Park, California 94025

Overview

SRI International's Artificial Intelligence Center currently has a number of projects in automatic planning. These range from formal work on the theoretical foundations of planning to investigations into real-world planning systems and real-time simulation. Several of these projects are summarized below. SIPE is an implemented system that further develops the approach pioneered by Sacerdoti in NOAH. Two formal approaches are described: one based on dynamic logic, the other on the situation calculus. Another project uses a formal approach to cooperative planning for multiple agents. The KAMP system plans natural-language utterances, combining a planning system with linguistic knowledge. Another project investigates execution-monitoring in real-world situations, as well as the inclusion of real-time simulation in the planning process. The last project described views planning more as an optimization problem as it deals with planning in an uncertain environment that includes hostile agents.

Domain-Independent Planning with Parallel Actions

David Wilkins

Our research involves domain-independent planners that are of particular interest since, in addition to planning techniques that are applicable in many domains, they provide a general planning capability. Two features found in many planning systems are also central to this work: hierarchical planning and parallel actions. Hierarchical planning is often necessary for real-world domains, because it helps avoid the tyranny of detail that would result from planning at the most primitive level. Parallel actions are also useful for real-world domains since such domains are often multiagent (e.g., they have two robot arms for constructing an object), and the best plans should employ these agents in parallel when possible.

Overview of SIPE

We have designed and implemented (in INTERLISP) a system, SIPE, (System for Interactive Planning and Execution Monitoring), that supports domain-independent planning. The program has produced correct parallel plans for problems in four different domains (the blocks world, cooking, aircraft operations, and a simple robotics assembly task). The system provides for hierarchical planning and parallel actions. Like most domain-independent planning systems, SIPE assumes discrete time, discrete states, and discrete operators. All relations mentioned in the world model are assumed to remain unchanged unless an action in the plan specifies that some relation has changed.

SIPE can generate plans automatically, but, unlike its predecessors, SIPE is designed to also enable interaction with users throughout the planning and plan-execution processes, if so desired. A plan is a set of partially ordered goals and actions composed by the system from operators (the system's description of actions it may perform). Because plans that do not achieve the desired goal may sometimes be generated, the system also has "critics" that find potential problems and attempt to correct them. In particular, most of the reasoning about interactions between parallel actions is done by the critics.

Representation

Invariant properties of objects in the domain are represented in a sort hierarchy, which allows inheritance of properties and the posting of constraints on the values of attributes of these objects. The relationships that change over time, and hence all goals, are represented in a version of first-order predicate calculus that is sorted and interacts with the knowledge in the sort hierarchy. Operators are represented in a perspicuous formalism developed by us in which the ability to post constraints on variables is a primary feature.

One of SIPE's most important advances over previous domain-independent planning systems is its ability to construct partial descriptions of unspecified objects. Constraints may place restrictions on the properties of an object (e.g., requiring certain attribute values for it in the sort hierarchy), and also require that certain relationships exist between an object and other objects (e.g., predicates that must be satisfied in a certain world state).

The formalism for representing operators in SIPE includes a means of specifying that some of the variables associated with an action or goal actually serve as resources for that action or goal. SIPE has specialized knowledge for handling resources. Mechanisms in the planning system automatically check for resource conflicts and ensure that these availability preconditions will be satisfied.

Operators allow the specification of purposes for determining plan rationale. Determining purposes correctly is necessary for correcting problematic parallel interactions, and for repairing a plan during execution. SIPE also allows the specification of deductive operators that deduce facts from the current world state. Besides simplifying operators, deductive operators are important in many domains for their ability to represent conditional effects. Deductive operators in SIPE may include both existential and universal quantifiers, thus providing a fairly rich formalism for deducing the effects (possibly conditional) of an action.

Parallel Interactions

As noted before, parallelism is considered beneficial because optimal plans in many domains require it. SIPE has new features and heuristics that aid in handling parallel interactions. These fall into four areas: (1) reasoning about resources, which is a major contribution of SIPE; (2) using constraints to generate correct parallel plans; (3) explicitly representing the purpose of each action and goal to help solve harmful interactions correctly; (4) taking advantage of helpful interactions.

Future Research

We intend to work towards a useful application of this work by extending SIPE to handle robotics assembly tasks. This will entail additional execution-monitoring capabilities and the use of information-gathering operators to incorporate conditional tests into the plans.

Contacts

SIPE was designed and implemented by Ann Robinson and David Wilkins. David Wilkins is principal investigator and should be contacted for further information. SIPE is described in more detail in [7]. David Wilkins and Stan Rosenschein are continuing the research on this project. The research reported here has been supported by Air Force Office of Scientific Research Contract No. F49620-79-C-0188.

Theoretical Foundations of Planning

Stan Rosenschein, Stuart Sheiber

Most AI planning research to date has been based on a simple state-transition model of action in which there is only one agent whose actions are always determinate. To handle complex domains realistically, we need to extend the model to allow actions with indeterminate outcomes (especially when outcomes differ in likelihood), as well as actions by more than one agent.

There are several ways of adding indeterminacy to the underlying framework. The situation calculus formulation of planning is able to express indeterminacy up to logical disjunction by simply having the axioms that express the effects of actions contain disjunctive postconditions. Unfortunately, the STRIPS formulation, which suppresses state variables and expresses the effects of actions as state-description transformations, is incapable of expressing this indeterminacy. Our work on dynamic-logic-based planning addresses this problem by combining some of the best features of STRIPS (e.g., the suppression of state variables and the use of structured search through a space of state descriptions) with the best features of the situation calculus (e.g., the possibility of disjunctive postconditions).

In a fairly straightforward extension of the dynamic-logic framework, certainty factors can be introduced into the model. By changing the formulas denoting truth or falsehood in the original logic to terms denoting probabilities in the new logic, we can preserve the essential character of the original approach while augmenting its expressive capabilities.

Concurrency can be included in the formal model by having the state transformations be parameterized by the actions of several agents instead of only one. If we are willing, in principle, to postulate a global state of the system (even though in practice we may have only incomplete knowledge of this state), we can conceive of the parallel execution of a primitive operation as being a single complex operation on the global state. Reasoning about sequences of actions by the various agents then involves reasoning about interleavings of primitive events. This would be difficult if the only way to perform this reasoning were to enumerate the combinatorially large number of execution sequences and examine each in turn. Fortunately, in typical domains the effects of an action by one agent are ordinarily invariant under most actions by other agents. This fact can be used to facilitate the reasoning.

Domain-independent planning formalisms are judged not only by their expressive capabilities but also by the ease with which domain-specific operators can be described. This is the source of the often heard objections to "frame axioms" in the situation calculus. In essence, the STRIPS assumption (i.e., that relations not mentioned in the operator description remain invariant) is quite challenging to formalize. Another approach would be to adopt syntactic conventions that could be used to compactly describe the intended model. In this view, when discussing the semantics of an operator, we would assume that "frame axioms" are in force, but they would not ordinarily be written out in full. Rather, they would be supplied uniformly by convention. Similarly, "isolation conditions" can help simplify descriptions of state transitions in the case of multiple agents.

The framework for this research is described in [6]. The research reported here has been supported by Air Force Office of Scientific Research Contract F49620-79-C-0188.

A Situational-Calculus Approach to Planning

Richard Waldinger, Zohar Manna (Stanford University, Weizmann Institute of Science)

In this project, planning and program synthesis are regarded as a form of theorem-proving in a situational calculus, i.e., a logical framework in which states are explicit objects. In the adopted version of situational calculus, plans are also objects and DO is a function of a plan and an initial state which produces a final state. The task of planning to achieve a desired condition is then transformed into one of proving the existence of a plan producing a final state in which the condition is true. The plan is extracted from the proof of its existence.

A special emphasis of the project is on the construction of plans with loops and branches. Although the formalism has been developed for sequential plans, an adaptation to concurrent planning is being contemplated. In the concurrent formalism, DO would be represented as a relation between a plan, an initial state, and a final state, rather than as a function.

Principal applications are to programming and common-sense reasoning. The approach is first-order and can be embedded in existing first-order theorem provers with induction.

This research is described in more detail in [4] and [5], and has been supported by NSF Grant MCS-78-02591, NSF Grant MCS-8105565, and the Office of Naval Research Contract No. N00014-75-C-0816.

Common-Knowledge Representations for Cooperative Planning

Kurt Konolige

While speech act theory shows promise as a general framework for reasoning about communication acts and integrating them with planning systems, it also requires significant resources for deduction. This arises because speech acts entail the recognition of intention, e.g., an utterance may mean something like "it is mutually believed that the hearer believes that the speaker wants the hearer to believe that the speaker believes that P," where P is some proposition about the world. It is then a nontrivial deduction to infer that the speaker actually does believe P.

Although in the context of general conversation the recognition of complex intentions is necessary for successful communication, it may be possible to derive simpler forms for use in structured cooperative-planning situations. We are investigating a model of cooperative planning in which there are two agents in a blocks-world environment who communicate to maintain a common knowledge base about the state of the physical world. This model permits a much simpler treatment of speech acts, while still exhibiting interesting interactions between communicating and planning.

This research, described in greater detail in [3], has been supported by the Office of Naval Research under Contract No. N00014-80-C-0296.

Natural-Language Generation Based on Planning

Doug Appelt, Barbara Grosz, Bob Moore

We are currently developing a theory of natural-language generation based on planning. The KAMP (Knowlege and Modalities Planner) system [1, 2] is designed to plan actions that affect the knowledge and intentions of multiple agents cooperating on a single task. KAMP is a multiagent hierarchical planner that refines plans that include abstractly specified communication actions, such as informing and requesting, into completely specified plans involving the utterance of specific English sentences. KAMP can reason about how a single utterance can satisfy several of a speaker's goals simultaneously and, in addition, how a speaker can combine physical and linguistic actions to communicate his intentions.

At present, two major current research efforts are (1) the development of more powerful formal tools for reasoning about agents' plans and intentions, and (2) expanding the linguistic capability of KAMP by the specification of detailed linguistic knowledge in a formalism that can be easily used by a planning system. Increased linguistic capability will enable us to investigate the planning of coherent discourse as well as the planning of single sentences.

The need to generate fluent natural language provides a motivation for investigating a number of issues in planning that are not directly related to language -- for example, the development of flexible techniques for handling blocked plans, the satisfying of mutually exclusive goals, and the commingling of planning and execution.

This research, described in more detail in [1] and [2], has been supported by NSF Grant MCS-8115105.

Adaptive Modeling and Real-time Simulation

Mabry Tyson, Tom Garvey, Art Farley

Decision-making in complex real-world situations is far more complicated than present planning systems can support. Of the many aspects of planning that need to be investigated, this project focuses on responding to a changing environment in which a plan is being executed.

Early planners usually concentrated on developing plans while ignoring their execution phase. The latter, in contrast, constitutes the main focus of our own work. We want to be able to adapt to unforeseen events that could interfere with both the present and future execution of the plan. For example, plan actions may not achieve the desired results, thereby disrupting future actions and possibly preventing the plan's goals from being achieved. Conditions existing at the time of plan creation may change, thereby invalidating assumptions used to satisfy preconditions of actions required by the plan. In particular, uncooperative or hostile agents may actively interfere with actions carried out as part of the plan.

If a plan is to achieve its goals, it cannot simply be initiated and forgotten. The plan must be monitored and important effects must be confirmed. If an action does not achieve its planned effect, alternative actions may be necessary. We allow for a plan that may have some monitoring of its actions built into it. Constraints on resource usage, perhaps to accomodate future demands upon those resources, must be obeyed. Violation of these constraints should be noted, as this may force a partial replanning. Such violations might be caused by external agents (or actions of other plans) and are therefore not likely to be adequately monitored as part of the plan itself.

If time is critical, obstructions to successful completion of a plan should be identified as soon as possible. Early detection of potential problems can lead to simpler modifications of the original plan (perhaps just a schedule change will suffice). Failure to detect problems until they actually stop the execution of an action may block the ultimate goal from ever being achieved. We are investigating the question of how to identify potential problems and the particular portions of the plan they may affect. The early detection of these problems will be facilitated by propagating backward through time the requirements and conditions of a plan's various stages. If a situation appears problematical, it may be simulated to determine whether or not the plan will be disrupted (thus simulating existing conditions forward through time).

In our view, a computer-based system to support decision-making in a changing real-world situation could monitor several plans simultaneously. Because each plan is composed of actions that take place over different periods of time, their temporal interrelations must be modeled properly. Information about the changing environment would be fed into the system. Some information might be received as a result of monitoring built into the plans. Information from other sources would also be integrated into the computer's model of the world. Planned actions and their effects would be merged into the model as they happen, but the system must be able to recover if it discovers subsequently that the effects had not actually been realized.

This system would be responsible for noting conditions that could interfere with planned actions. Information from a plan's self-monitoring or other sources might indicate possible problems. In certain simple instances the system may be able to solve the problem by altering the schedule of actions in the plan. More complicated replanning would probably require a second pass through the mechanism by which the original plan was generated. Since some of the plan would have already been executed and since more information is now known, the second pass would probably be easier but also more time-critical. The system should be able to accept a modified plan to replace an existing plan.

Building such a system is beyond the scope of the present project. We are merely investigating the requirements for such a system and the manner in which it could actually be implemented.

The research reported here has been supported by RADC Contract No. F30602-81-C-0218.

Real-time Planning in Conflict Situations

Tom Garvey, Peter Cheeseman

We have been investigating extensions of current planning paradigms in order to allow the creation of strategies for self-defense in conflict situations. This requires an ability to perform imprecise inferencing over potentially large amounts of uncertain information, rapidly enough to provide a valid and timely response. Difficulties arise from a number of directions:

(1) Uncertainty about the status, composition, and location of your own resources.

(2) Uncertainty about the intent, capabilities, and disposition of your opponent's resources.

(3) Operators that are non-deterministic and, once initiated, require finite time to complete.

(4) Activities of hostile opponents.

(5) "Entropic" increase in uncertainty -- that is, there are ongoing, unspecified processes that have the ultimate effect of degrading the quality of certain of our information (thereby increasing our effective entropy), purely as a function of the age of the information.

Work to date has focussed on development of representations and paradigms -- implementation is at a very early stage. We have developed structured representations for situations that incorporate interrelations among objects and actors with associated likelihoods. These models allow us to determine our current view of the situation, critical gaps in information, and likely extrapolations of the situation. We have associated certain operators with process models that prescribe the various states of the process, the possible transitions, the conditions that must prevail (or be maintained) while the process is in a state, and likelihoods of each transition. We have taken a view of planning as a process whose intent is to optimize the chances of achieving stated goals. In particular, the planner may wish to maximize the probability of surviving the conflict, while simultaneously minimizing the resources expended in the operation. The approach we are pursuing to accomplish this has several steps:

(1) Generate a partial plan (tree structure) of possibilities, with probabilities associated with each branch.

(2) Select the best branch for execution or further planning.

(3) Initiate execution of the first step and monitor the operation.

(4) Update the plan, and continue until goal is attained.

We relate our information needs (as determined from situation models) to our acquisition capabilities, in order to determine ways of acquiring adequate situation data. Estimates of hostile intent are presently limited to detecting an actual attack. Possible instantiations of hostile actors in the scenario are suggested by models; each actor has an associated process model that is used to determine an appropriate response. A hierarchical approach is necessary here in order to limit branching at a detailed level. Several possible responses and contingencies may then be elaborated and combined into a Markov tree with associated branch probabilities. There are a number of important aspects to this approach. The use of hierarchical planning enables a strategy to be developed, while deferring details to a later time. This should enable us to handle the inherent large branching factors of real-world plans in a computationally efficient manner. The shift from attempting to arrive at a goal state to attempting to optimize a set of parameters offers the potential for effectively representing and using operators and states that are continuous and fuzzy. The use of process models explicitly introduces the effects of operations over time, and simultaneously provides a directed indexing scheme for conceiving responses to an opponents actions. We are developing a small implementation to experiment with the concepts outlined here.

The research reported here has been supported by the Office of Naval Research under Contract No. N0014-81-C-0115.

References

1. Appelt, D. E., "Planning Natural Language Utterances to Satisfy Multiple Goals," SRI Artificial Intelligence Center, Technical Note 259, 1982.
2. Appelt, D. E., "Planning Natural Language Utterances," Proceedings of the National Conference on Artificial Intelligence, 1982.
3. Konolige, K., "A First-Order Formalization of Knowledge and Action for a Multiagent Planning System", SRI Artificial Intelligence Center, Technical Note 232, 1980.
4. Manna, Z., and Waldinger, R., "A Deductive Approach to Program Synthesis," ACM Transactions on Programming Languages and Systems, Vol. 2, No. 1 (January 1980), pp. 92-121.
5. Manna, Z., and Waldinger, R., "Problematic Features of Programming Languages: A Situational-Calculus Approach," Acta Informatica, Vol. 16 (1981), pp. 371-426.
6. Rosenschein, S., "Plan Synthesis: A Logical Perspective", Proceedings IJCAI--81, Vancouver, British Columbia, 1981, pp. 331-337.
7. Wilkins, D., "Domain Independent Planning: Representation and Plan Generation", SRI International Artificial Intelligence Center, Technical Note 266, August 1982.

Design of Integrated Circuits (KBVLSI)

Xerox PARC: Dan Bobrow, Alan Bell,
Lynn Conway, Sanjay Mittal,
Mark Stefik

Stanford University: Harold Brown,
Gordon Foyster, Christopher Tong,
Narinder Singh

Fairchild AI Lab: Harry Barrow

The general research interest of this project is design. The project (the KBVLSI project) is focussed on the design of integrated circuits. One of its goals is the creation of a system (called Palladio) to be an expert assistant for a community of integrated-circuit designers. Another goal for this project is to develop methods for such communities to collaborate and compete in the design of expert systems. We are working to make it possible for designers to design not only circuits, but also to articulate the knowledge required for designing circuits.

The project crosses three institutions and there is a diversity of interests among the researchers. Topics of current interest include:

- □ Theoretical and pragmatic frameworks for engineering knowledge.
- □ Theory and measurement of abstraction levels.
- □ Development and experimentation with languages for knowledge representation.
- □ Knowledge representation and programming paradigms.
- □ Representation of goals and tradeoffs in design.
- □ Representation and use of design alternatives.
- □ Representation and problem solving using constraints.
- □ Multiple level simulation and symbolic execution.
- □ Issues in the design of community knowledge bases.
- □ Development of pictorial descriptions.
- □ Techniques for knowledge compilation.

We are using the LOOPS language developed by Dan Bobrow and Mark Stefik. LOOPS is a programming system that supports object-oriented programming, data-oriented programming, and rule-oriented programming and integrates them with procedure-oriented programming in Interlisp-D.

The project has been underway for just over two years. Its current status can be summarized as follows:

1. The LOOPS language has been substantially developed, and has been in use for over a year.
2. Some theoretical work on the creation of suitable levels of abstraction for designing circuits has been reported.
3. There is a design editor and simulator for our CPS (clocked primiitve switches) level of representation.
4. There is an editor and simulator for the CRL (clocked register and logic) level of representation.
5. Bits and pieces of editors and representations of two other levels of abstraction have been completed.
6. We are currently experimenting with our rule language, and with programs that reason about parts of circuit design.
7. We are beginning to prepare some experiments on "the engineering of knowledge" in a way that will admit the participation of a community of designers.
8. Substantial progress has been made on two doctoral theses by students on the project.

References

1. Bobrow, D. G., Stefik, M. The LOOPS Manual: A Data, Object, and Rule-Oriented Programming System for Interlisp, Knowledge-Based VLSI Design Group, Memo KB-VLSI-81-13 (working paper) August, 1982.
2. Bobrow, D. G., Stefik, M. LOOPS -- Data and Object Oriented Programming for Interlisp. European Conference on Artificial Intelligence, July, 1982.
3. Brown, H., & Stefik, M. Palladio: An expert Assistant for Integrated Circuit Design, Knowledge-Based VLSI Design Group, Memo KB-VLSI-82-17, April, 1982.
4. Stefik, M. & Conway, L. "Towards the principled engineering of knowledge", The AI Magazine, Vol III, Summer, 1982.
5. Stefik, M., Bobrow, D. G., Bell, A., Brown, H., Conway, L., and Tong, C. "The partitioning of concerns in digital design", Proceedings of the Conference on Advanced Research in VLSI, Penfield, P. (Ed), January 25-27, 1982.