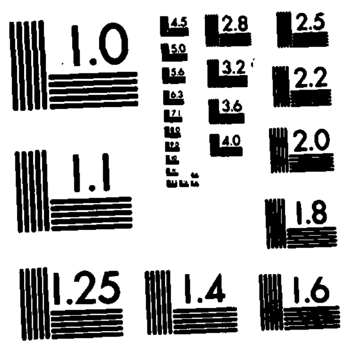
AD-A125 192 RESEARCH ON PARALLELISM IN PROBLEM-SOLVING SYSTEMS(U) 1/1
SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE
CENTER D E WILKINS 03 SEP 82 AFOSR-TR-83-0035
UNCLASSIFIED F49620-79-C-0188 F/G 12/1 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

3

AD A125192

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 83-0035	2. GOVT ACCESSION NO. AD-A125 192	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RESEARCH ON PARALLELISM IN PROBLEM-SOLVING SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED INTERIM, 1 Sep 81-31 Aug 82
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David E. Wilkins		8. CONTRACT OR GRANT NUMBER(s) F49620-79-C-0188
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Center, Computer Science and Technology Division, SRI International, 333 Ravenswood, Menlo Park CA 94025		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE61102F; 2304/A2
11. CONTROLLING OFFICE NAME AND ADDRESS Mathematical & Information Sciences Directorate Air Force Office of Scientific Research Bolling AFB DC 20332		12. REPORT DATE 3 Sep 82
		13. NUMBER OF PAGES 9
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DTIC ELECTE MAR 3 1983 S D B

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report summarizes SRI International's third year of research on a system for automatically generating hierarchical plans containing parallel (concurrent) actions. This is a general planning and problem-solving system that is not tied to a particular domain. Results of this research might eventually be used in the development of systems for automatically generating plans to coordinate the activities of military personnel engaged in a common mission. Other domains of application could include logistical planning, planning the concurrent use of many computers on a single network, and planning the movements of (CONTINUED)

DD FORM 1 JAN 73 1473

DTIC FILE COPY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ITEM #20, CONTINUED: a robot arm.

Sacerdoti's NOAH [2] is one of the most important general planners. In the first year of this project, as was described in detail in the first annual report, the researchers produced an analysis of NOAH's shortcomings along with some ideas for improvement, a design of a general planning system that advances the state of the art while eliminating these shortcomings, and the beginning of an implementation of the designed system in a computer program. Some of the results of the new design appeared in a technical publication in the 1980 Proceedings of the American Association for Artificial Intelligence.

During the second year, all parts of the system were designed in detail and the implementation was made more complete. Extensions of previous research include a formalism for describing actions, the ability to reason about resources, the ability to partially describe objects, the use of deduction for determining the effects of actions, and the recognition of problems and fortuitous side effects in parallel plans. These extensions, which were summarized in the second annual report, are described in some detail in [5].

In this third project year, the research has been concerned with theoretical foundations of planning, the use of metaplanning and higher-level strategies, with further automation of such system features as resource allocation, constraint satisfaction, and operator representation capabilities, and with an initial investigation of execution monitoring. In addition a major paper was prepared describing the research that has been done on this project. That paper is being submitted to the Artificial Intelligence Journal (SRI Artificial Intelligence Center Technical Note 266). This report contains only brief summaries of the research accomplishments of the past year.

Accession For		
NTIS GRA&I	☑	
DTIC TAB	☐	
Unannounced	☐	
Justification		
By		
Distribution/		
Availability Codes		
Dist	Avail and/or Special	
A		

DTIC COPY INSPECTED 2

SRI International

AFOSR-TR- 83-0035

RESEARCH ON PARALLELISM IN PROBLEM-SOLVING SYSTEMS

September 3, 1982

Third Annual Technical Report

SRI Project 8871

By: David E. Wilkins
Computer Scientist

Artificial Intelligence Center
Computer Science and Technology Division

Prepared for:

Air Force Office of Scientific Research
Building 410
Bolling Air Force Base
Washington, D.C. 20332

Attention: Captain William Price
Contract No. F49620-79-C-0188

Approved for public release; distribution unlimited.

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 859-6200 • TWX: 910-373-2046 • Telex: 334 486

023 080

1. Introduction

This report summarizes SRI International's third year of research on a system for automatically generating hierarchical plans containing parallel (concurrent) actions. This is a general planning and problem-solving system that is not tied to a particular domain. Results of this research might eventually be used in the development of systems for automatically generating plans to coordinate the activities of military personnel engaged in a common mission. Other domains of application could include logistical planning, planning the concurrent use of many computers on a single network, and planning the movements of a robot arm.

Sacerdoti's NOAH [2] is one of the most important general planners. In the first year of this project, as was described in detail in the first annual report, we produced an analysis of NOAH's shortcomings along with some ideas for improvement, a design of a general planning system that advances the state of the art while eliminating these shortcomings, and the beginning of an implementation of the designed system in a computer program. Some of the results of the new design appeared in a technical publication in the 1980 Proceedings of the American Association for Artificial Intelligence.

During the second year, all parts of the system were designed in detail and the implementation was made more complete. Extensions of previous research include a formalism for describing actions, the ability to reason about resources, the ability to partially describe objects, the use of deduction for determining the effects of actions, and the recognition of problems and fortuitous side effects in parallel plans. These extensions, which were summarized in the second annual report, are described in some detail in [5] (a copy of which is enclosed).

In this third project year, our research has been concerned with theoretical foundations of planning, the use of metaplanning and higher-level strategies, with further automation of such system features as resource allocation, constraint satisfaction, and operator representation capabilities, and with an initial investigation of execution monitoring. In addition a major paper was prepared describing the research that has been done on this project. That paper is being submitted to the Artificial Intelligence Journal and is enclosed herewith (SRI International

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DTIC
This technical report has been reviewed and is
approved for public release IAW AFR 190-12.
Distribution [illegible]limited.
MATTHEW J. KERPER
Chief, Technical Information Division

Artificial Intelligence Center Technical Note 266) [5]. This report contains only brief summaries of the research accomplishments of the past year; the enclosed paper provides more detail.

2. Extensions of Previous Research

We designed and implemented a system, SIPE, (System for Interactive Planning and Execution Monitoring), that incorporates the planning ideas developed during this project. Developments described in the first three areas below have been implemented in this program. During the past year we have been testing the program in different domains (see [5]) and have made progress in the following areas (which are described in more detail below):

- More flexible and uniform representations for objects and actions have been developed.
- New techniques have been developed for efficiently detecting and remedying harmful parallel interactions. The most important of these techniques is reasoning about resources.
- Primitive methods for enhancing the execution monitoring ability of SIPE have been developed.
- Metaplanning and its implications for domain independent planning have been investigated, resulting in a better understanding of some aspects of metaplanning.
- Theoretical foundations were developed for AI planning in domains involving concurrent action and operators with uncertain effects.

2.1 Representation of Actions and Objects

The planning representation problem involves representing the domain, goals, and operators. Operators are the system's representation of actions that may be performed in the domain or, in the hierarchical case, abstractions of actions that can be performed in the domain. An operator includes a description of how each action changes the state of the world. The formalism used for representation in SIPE was developed in the second year of the project (and is described in Technical Note 266), but was extended during the past year. Efficient constraint satisfaction and resource allocation routines were implemented to increase the power of the system. Two important capabilities were added to the formalism; these are described below.

The first new development is the ability to have loops inside operators. A variable is now permitted to be instantiated to a list of objects. The plot of the operator (i.e., the operator's instructions for accomplishing its goal) may then contain an ITERATE-BEGIN and an ITERATE-END, and the plot within these tokens will be reproduced in the resultant procedural net once for each different object in the list that is the instantiation of the iteration variable.

The second new development is the addition of new constraints that are understood by SIPE. These are the OPTIONAL-SAME and OPTIONAL-NOT-SAME constraints, which, like the SAME and NOT-SAME constraints, specify that a variable must be instantiated to be the same as (or not the same as) another variable. In the OPTIONAL case, however, the constraint is not binding. SIPE will attempt to satisfy it, but, if it cannot, planning will continue. This is useful for trying to avoid resource conflicts in parallel plans. It is nice to avoid parallel conflicts, but, if adequate resources are not available to accomplish the goals in parallel, then SIPE will assign the same resource and worry later about linearizing the parallel actions to avoid the conflict.

2.2 Reasoning About Parallel Interactions

SIPE recognizes both helpful and harmful parallel interactions, and has new features and heuristics that aid in handling them. These fall into four areas (the first three of which have been implemented during the past year): (1) reasoning about resources, which is the major contribution of SIPE; (2) using constraints to generate correct parallel plans; (4) taking advantage of helpful interactions; (3) explicitly representing the purpose of each action and goal to help solve harmful interactions correctly,

The use of OPTIONAL constraints to generate correct plans by avoiding resource conflicts has already been mentioned. SIPE recognizes helpful interactions and will try to order the plan further to take advantage of them. If a goal that must be made true on one parallel branch is actually made true on another parallel branch, the system will under certain conditions order the plan so that the other branch occurs first (if this causes no other conflicts). NOAH was not able to take advantage of such helpful effects. This is an important ability in many real-world domains, since helpful side effects occur frequently. For example, if parallel actions in a robot world both require the same tool, only one branch need plan to get the tool out of the tool box; the other branch should be able to recognize that the tool is already out on the table.

SIPE has specialized knowledge for handling resources; declaration of a resource associated with an action is a way of saying that one precondition of the action is that the resource be available. Mechanisms in the planning system, as they allocate and deallocate resources, automatically check for resource conflicts and ensure that these availability preconditions will be satisfied. One advantage of resources, therefore, is that they help in the axiomatization and representation of domains. The user of the planning system does not have to axiomatize as a precondition the availability of resources in the domain operators. (Such an axiomatization may be difficult, since the critics must use the representation correctly to recognize problems with unavailable resources.) This enables both SIPE's operators and plans to be shorter and easier to understand than similar operators and plans in domain independent parallel planning systems, such as NOAH and NONLIN [3]. Another important advantage of resources is that they help in early detection of problematic interactions on parallel branches. The system does not allow one branch to use an object that is a resource in a parallel branch. (This is described in detail in [5].)

2.3 Execution Monitoring

In real-world domains, things do not always proceed as planned. Therefore, it is desirable to develop better execution-monitoring techniques and better capabilities to replan when things do not go as expected. In complex domains it becomes increasingly important to use as much as possible of the old plan, rather than start all over when things go wrong. SIPE has addressed only some of the problems of execution monitoring; research is continuing in this area.

During execution of a plan in SIPE, some person or computer system monitoring the execution can specify what actions have been performed and what changes have occurred in the domain being modeled. In accordance with this, the plan can be updated interactively to cope with unanticipated occurrences. Planning and plan execution can be intermixed by producing a plan for part of an activity and then executing some or all of that plan before elaborating on the remaining portion.

At any point in the plan, the user can inform the system of a predicate that is now true (though SIPE may have thought it was false). The program will look through the plan and find all goals that are affected by this new predicate. Since SIPE understands the rationale of nodes in the plan (through purposes), it can determine how changes affect the plan. For example, if a later purpose is suddenly accomplished unexpectedly, SIPE can notice the helpful effect and eliminate a whole section of the plan because it knows the preparatory steps are only there to accomplish the purpose. If an unexpected event causes a problem, the system will suggest all the solutions it can find. SIPE's repertoire of techniques for finding such solutions is not very sophisticated, however. It includes: (1) instantiating a variable differently (e.g., using a different resource if something has gone wrong with the one originally used in the plan), (2) finding relevant operators to accomplish a goal that is no longer true (and inserting the new subplan correctly in the original plan), and (3) finding a higher level from which to replan if the problems are widespread.

2.4 Metaplanning

Despite the considerable amount of discussion about metaplanning in the artificial intelligence literature, no domain independent planner has done interesting metaplanning. Research on SIPE reveals the cause of this. First, the very concept of meta-planning had to be clarified because the term is used so vaguely by so many. This is done in [5]. One obstacle to greater precision with regard to metaplanning is that there is often no clear dividing line between the external domain and the planning process (contrary to Wilensky's argument in [4]). Given any particular system, it will likely be obvious what is at a metalevel in that system. But any particular piece of knowledge might be encoded at either the metalevel or the domain level – and it is not always clear which is best. It is trivial to convert any domain operator into a metaoperator, and many metaoperators can probably be wired into the domain at the time the domain representation is being designed. Higher-level domain knowledge generally blends into search control knowledge.

There are several domain independent planners in the literature, but none of them does interesting metaplanning. There is good reason for this, as these domain independent formalisms are simply not adequate for expressing interesting metaknowledge. Systems such as SIPE that do hierarchical planning can use abstract operators to encode some metaplanning knowledge, but the most interesting metaplanning ideas cannot be encoding in this manner because these formalisms generally use a *model approach* for representing the domain. The essential characteristic of this approach is that all relationships that hold in the domain are expressed directly (e.g., disjunctions are geneally not allowed), in the sense that the model can be queried in a lookup manner to return an answer quickly about the truth value of a relationship. This efficient querying ability is, of course, the motivation for the model approach. The disadvantage of the model approach is that there are many things that cannot be represented because they do not admit of such direct representation. In particular, what we need to say about plans at a metalevel cannot fit easily into the model approach, since it will not be reasonable to represent explicitly (for example) every property of a plan, a failed search branch, an operator, or a constraint that we might want to reason about at the metalevel. Thus, a reasonable language for metaplanning must be richer than those commonly used in domain independent planners.

2.5 Theoretical Foundations

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 where outcomes differ in likelihood) as well as actions by more than one agent. This research was carried out independently from the work on SIPE, and is not described in [5].

There are several ways of adding indeterminacy to the underlying framework. The situation calculus formulation of planning [1] is able to express indeterminacy up to logical disjunction by simply having the axioms that express the effects of an action 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 extending its expressive capabilities.

The inclusion of concurrency in the formal model can be handled 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 mechanisms 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.

3. PUBLICATIONS AND CONFERENCES

This project supported in part the attendance of David Wilkins at the Canadian Society for Computational Studies of Intelligence Conference at the University of Saskatchewan, 17-19 May 1982. He delivered a paper, entitled "Parallelism in Planning and Problem Solving: Reasoning About Resources", which describes research performed on this project. The paper is available in the conference proceedings.

As stated previously, SRI Tech Note 166, entitled "Domain Independent Planning: Representation and Plan Generation", describes research done under this project. First published in August 1982, it is currently being submitted to the Artificial Intelligence Journal. It is enclosed with this report.

REFERENCES

1. McCarthy, J., and Hayes, P. "Some Philosophical Problems from the Standpoint of Artificial Intelligence", *Readings in Artificial Intelligence*, Nilsson and Webber, ed., Tioga Publishing, Palo Alto, California, 1981, pp. 431–450.

2. Sacerdoti, E., *A Structure for Plans and Behavior*, Elsevier, New York, 1977.

3. Tate, A., "Generating Project Networks", *Proceedings IJCAI-77*, Cambridge, Massachusetts, 1977, pp. 888-893.

4. Wilensky, R., "Meta-planning", *Proceedings AAAI-80*, Stanford, California, 1980, pp. 334-336.

5. Wilkins, D., "Domain Independent Planning: Representation and Plan Generation", Technical Note 266, Artificial Intelligence Center, SRI International, Menlo Park, California, August 1982.

END