(2)

RADC-TR-90-404, Vol I (of 18)
Final Technical Report
December 1990

EXECUTIVE SUMMARY

Northeast Artificial Intelligence Consortium (NAIC)

Volker Weiss and James F. Brulé

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

This effort was funded partially by the Laboratory Director's fund.

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the RADC Public Affairs Division (PA) and is releasable to the National Technical Information Services (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-90-404, Volume I (of 18) has been reviewed and is approved for publication.

APPROVED:

SHARON M. WALTER
Project Engineer

APPROVED:

RAYMOND P. URTZ, JR.
Technical Director
Directorate of Command & Control

FOR THE COMMANDER:

BILLY G. OAKS
Directorate of Plans & Programs

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
December 1990

3. REPORT TYPE AND DATES COVERED
Final Sep 84 - Dec 89

4. TITLE AND SUBTITLE
EXECUTIVE SUMMARY

5. FUNDING NUMBERS
C - F30602-85-C-0008
PE - 62702F
PR - 5581
TA - 27
WU - 13
(See reverse)

6. AUTHOR(S)
Volker Weiss and James F. Brulé

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Northeast Artificial Intelligence Consortium (NAIC)
Science & Technology Center, Rm 2-296
111 College Place, Syracuse University
Syracuse NY 13244-4100

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Air Development Center (COES)
Griffiss AFB NY 13441-5700

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RADC-TR-90-404, Vol I (of 18)

11. SUPPLEMENTARY NOTES
RADC Project Engineer: Sharon Walter/COES/(315) 330-3577
This effort was funded partially by the Laboratory Director's fund.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
The Northeast Artificial Intelligence Consortium (NAIC) was created by the Air Force Systems Command, Rome Air Development Center, and the Office of Scientific Research. Its purpose was to conduct pertinent research in artificial intelligence and to perform activities ancillary to this research. This report describes progress during the existence of the NAIC on the technical research tasks undertaken at the member universities. The topics covered in general are: versatile expert system for equipment maintenance, distributed AI for communications system control, automatic photointerpretation, time-oriented problem solving, speech understanding systems, knowledge base maintenance, hardware architectures for very large systems, knowledge-based reasoning and planning, and a knowledge acquisition, assistance, and explanation system.

This volume provides the Executive Summary of the NAIC.

14. SUBJECT TERMS Artificial Intelligence, Expert Systems, Distributed AI, Problem Solving, Speech Understanding, Machine Vision, Knowledge Based Systems, Logic Programming, Logic Programming,

15. NUMBER OF PAGES
76

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Block 5 (Cont'd) Funding Numbers

PE - 62702F	PE - 61102F	PE - 61102F	PE - 33126F	PE - 61101F
PR - 5581	PR - 2304	PR - 2304	PR - 2155	PR - LDFP
TA - 27	TA - J5	TA - J5	TA - 02	TA - 27
WU - 23	WU - 01	WU - 15	WU - 10	WU - 01

Block 14 (Cont'd) Subject Terms

Planning, Knowledge Acquisition.

1.1 INTRODUCTION

1.1.1 The Northeast Artificial Intelligence Consortium

This is the final report of the Northeast Artificial Intelligence Consortium (NAIC), a collaboration of eight institutions of higher learning organized for the purpose of developing research and education in artificial intelligence (AI) in the northeastern area of the United States. The participating institutions were:

State University of New York at Buffalo, Buffalo NY
Clarkson University, Potsdam NY
Colgate University, Hamilton NY
The University of Massachusetts at Amherst, Amherst MA
Rensselaer Polytechnic Institute, Troy NY
The University of Rochester, Rochester NY
Rochester Institute of Technology, Rochester NY
Syracuse University, Syracuse NY

1.1.2 Objectives of the Consortium

The Consortium's *raison d'etre* was to advance the state of research and education in AI in the northeastern United States. The plan for achieving this objective was twofold: first, to sponsor specific research tasks which by their nature would advance the state of research capabilities within the northeast. Second, through the funding of a management effort which would oversee all the individual projects and foster communication between the members of each, it was hoped that collaborative efforts would emerge, and various side effects would be generated in service of the overall objectives. These plans have proved successful in producing a Consortium which has achieved both regional and national recognition, and which has indeed advanced the state of AI expertise in the northeast.

Over the course of the five years, nineteen research projects were funded, of which four were collaborative efforts between as many as four institutions. Consortium-wide meetings were held two to three times per year, drawing together the diverse resources offered by each school in an atmosphere of collegiality and professional exchange. Seminars were presented at each campus on a variety of topics, satellite workshops were sponsored, and workshops of national stature were supported. Attention was drawn to the Consortium through its co-sponsorship of activities by national (AAAI, IEEE) and regional (Association for Intelligent Systems Technology, local IEEE chapters) organizations, and a live satellite teleconference introducing the NAIC was broadcast to IBM sites across the eastern seaboard.

A technical report series was instituted and grew to include over 70 titles, active participation with industry was achieved (particularly with Texas Instruments and Rockwell), and new relationships with similar organizations were forged and maintained. Money from new contracts was integrated into the Consortium, and in the height of collaboration, a project involving researchers at four schools was conducted.

Researchers at each institution possess their own expertise and interests, and addressed a varied group of problems in AI that were of interest to the Air Force and beyond. Where in the beginning virtually all of these tasks had been independent, a quarter of them were collaborative in a formal sense. In an informal sense, each project was the beneficiary of the input and observations of other Consortium researchers, gleaned from annual and topical meetings. Reports of their accomplishments are included in the volumes that follow this one; executive summaries are included later in this volume.

1.1.3 Research Projects

The topics under study and the Principal Investigators ("PIs") at each institution are:

DISCUSSING, USING, AND RECOGNIZING PLANS

PIs: Stuart C. Shapiro (SUNY Buffalo)
Beverly Woolf (University of Massachusetts - Amherst)
(Volume 2)

VMES: A NETWORK-BASED VERSATILE MAINTENANCE EXPERT SYSTEM

PIs: Sargur N. Srihari, Stuart C. Shapiro
State University of New York at Buffalo
(Volume 3)

DISTRIBUTED AI FOR COMMUNICATION SYSTEMS CONTROL

PIs: Robert Meyer
Susan Conry
Clarkson University
(Volume 4)

DISTRIBUTED PROBLEM SOLVING FOR DYNAMIC PLANNING IN THE PRESENCE OF TIME CONSTRAINTS

PIs: Susan E. Conry, Robert A. Meyer (Clarkson University)
Victor Lesser, Paul Cohen (University of Massachusetts)
(Volume 5)

BUILDING AN INTELLIGENT ASSISTANT: THE ACQUISITION, INTEGRATION, AND MAINTENANCE OF COMPLEX DISTRIBUTED TASKS

PIs: Victor Lesser, W. Bruce Croft, and Beverly Woolf
The University of Massachusetts at Amherst
(Volume 6)

AUTOMATIC PHOTOINTERPRETATION

PIs: James Modestino
Arthur Sanderson
Rennselaer Polytechnic Institute
(Volume 7)

ARTIFICIAL INTELLIGENCE APPLICATIONS TO SPEECH RECOGNITION

PIs: Harvey Rhody, John A. Biles
Rochester Institute of Technology
(Volume 8)

PLANNER SYSTEM FOR THE APPLICATION OF INDICATIONS AND WARNING

PI: Sergei Nirenburg
Colgate University
(Volume 9)

TIME-ORIENTED PROBLEM SOLVING

PI: James F. Allen
The University of Rochester
(Volume 10)

INFERENCE TECHNIQUES FOR KNOWLEDGE BASE MAINTENANCE USING LOGIC PROGRAMMING METHODOLOGIES

PI: Kenneth A. Bowen
Syracuse University
(Volume 11)

COMPUTER ARCHITECTURES FOR VERY LARGE KNOWLEDGE BASES

PI: P. Bruce Berra
Syracuse University
(Volume 12)

PARALLEL VISION SYSTEMS; IMAGE UNDERSTANDING AND INTELLIGENT PARALLEL SYSTEMS

PIs: Christopher Brown, Randal Nelson
University of Rochester
(Volume 13)

KNOWLEDGE BASE RETRIEVAL USING PLAUSIBLE INFERENCE

PIs: Bruce Croft, Paul Cohen
University of Massachusetts - Amherst
(Volume 14)

STRATEGIES FOR COUPLING SYMBOLIC AND NUMERICAL COMPUTATION IN KNOWLEDGE BASED SYSTEMS

PI: Minsoo Suk
Syracuse University
(Volume 15)

INTELLIGENT SIGNAL PROCESSING FOR SURVEILLANCE SYSTEMS

PIs: Harvey Rhody (Rochester Institute of Technology)
James Modestinc (Rensselaer Polytechnic Institute)
David Sher (SUNY Buffalo)
Christopher Brown (University of Rochester) *consultant*
(Volume 16)

A KNOWLEDGE ACQUISITION, SYSTEMS, AND EXPLANATION SYSTEM: HIGH-LEVEL ADAPTIVE SIGNAL PROCESSING

PIs: Victor Lesser (University of Massachusetts - Amherst)
Hamid Nawab (Boston University)
(Volume 17)

A DISTRIBUTED ARTIFICIAL INTELLIGENCE APPROACH TO INFORMATION FUSION AND OBJECT CLASSIFICATION

PI: Pramod Varshney
Syracuse University
(Volume 18)

1.2. MANAGEMENT STRUCTURE

The original Director of the NAIC was Dr. Bradley Strait, of Syracuse University; the original Manager was Dr. Robert Cotellessa, of Clarkson University. Over the first two years of the Consortium, the management structure evolved from a loosely-structured "contract management" approach into its present form: a coherently structured confederation of collaborating researchers and institutions. This evolution became most noticeable when the current Director, Dr. Volker Weiss, initiated an Executive Committee after assuming the role of Director in 1986. This seminal development led to the creation of the position of Managing Director, which was filled after a national search in 1987. Since that time, the management structure has remained stable, and may be described as follows:

The Director is the principal agent of the prime contractor (Syracuse University). The Executive Committee is composed of four representatives from member institutions, each serving on the Committee for two years on a staggered schedule. The Executive Committee advises the Director on long-range plans and policies;

together, the Executive Committee and the Director set the course for the NAIC to follow.

The Managing Director (Mr. James Brulé) reports to the Director and the Executive Committee; his responsibilities are to implement the plans formulated by the Executive Committee on behalf of the Consortium, manage the day-to-day operations of the Consortium, and maintain administrative liaisons between member institutions, RADC, and administrative bodies within Syracuse University relative to the Consortium. An Administrative Secretary (Ms. Jeanette Fernandes) reports to the Managing Director, and along with the Managing Director forms the full-time administrative component of the Consortium. This component prepares reports, organizes NAIC meetings and briefings, aids in the establishment of committees and advisory boards, facilitates the electronic networking of Consortium members, arranges vendor presentations, organizes educational efforts, and represents the Consortium at an administrative level to other universities, funding sources, and the commercial sector.

Colgate University terminated its participation in the NAIC when its principal investigator, Dr. Sergei Nirenburg, accepted a position at Carnegie Mellon University. Conversely, Boston University became a *de facto* member when one of its faculty, Dr. Hamid Nawab, entered into collaborative research with Dr. Lesser of the University of Massachusetts.

1.2.1 Intra-School Representatives

The Principal Investigators at each institution are responsible for both the technical and ancillary functions at their respective institutions. The Principal Investigators and researchers involved in the projects at each institution are:

Stuart C. Shapiro
Sargur N. Srihari
David Sher
Shambhu J. Upadhyaya
State University of New York at Buffalo

Susan Conry
Robert Meyer
Clarkson University

Sergei Nirenburg
Colgate University

Victor Lesser
Paul Cohen
W. Bruce Croft
Beverly Woolf
University of Massachusetts - Amherst

Arthur Sanderson
James W. Modestino
Rensselaer Polytechnic Institute

James F. Allen
Christopher Brown
Randal Nelson
Henry Kautz
Josh Tenenberg
University of Rochester

Harvey Rhody
John Biles
Rochester Institute of Technology

P. Bruce Berra
Kenneth A. Bowen
Minsoo Suk
Pramod Varshney
Howard Blair
Syracuse University

Hamid Nawab
Boston University

1.3 TECHNICAL TASKS

Detailed descriptions of research tasks under investigation at each of the member institutions of the Consortium are found in subsequent volumes. Short descriptions of the research at each institution follow.

1.3.1 Discussing, Using, and Recognizing Plans (SUNY-UMass): Vol. 2

This project, also know as the Natural Language Planning Project, has been a joint project of a research group at SUNY at Buffalo (UB), led by Dr. Stuart Shapiro, and a research group at the University of Massachusetts at Amherst (UMass), led by Dr. Beverly Woolf. The project was devoted to the investigation of a knowledge representation design compatible with the intensional knowledge representation theory previously developed by Dr. Shapiro and his co-workers, and capable of providing a natural language interacting system with the ability to discuss, use, and recognize plans. The UB group was responsible for: the development, improvement, and maintenance of the knowledge representation, reasoning, and natural language processing software to be used in the project; for developing a representation of plans and associated concepts; for developing basic techniques for discussing plans in English; and for plan recognition. The UMass group was responsible for analyzing the chosen domains, principally the domain of tutoring interactions, for testing the developments of the Buffalo group by trying to apply them to these domains, and for suggesting changes to the representing of plans. With the support of the NAIC and of Texas Instruments, both groups used TI Explorers to do their work.

1.3.1.1 TECHNICAL PROGRESS

The objectives of this project were to:

1. design a representation for plans and rules for reasoning about plans within an established knowledge representation/reasoning (KRR) system; enhance the KRR system so that it can act according to such plans;

2. write a grammar to direct an established natural language processing (NLP) system to analyze English sentences about plans and represent the semantic/conceptual content of the sentences in the representation designed for objective (1); the resulting NLP system should be able to: accept sentences describing plans, and add the plans to its "plan library"; answer questions about the plans in its plan library; accept sentences

describing the actions of others; and recognize when those actions constitute the carrying out of a plan in its plan library.

The KRR system used was SNePS, and the NLP system to be modified for this purpose was CASSIE. The UB group was responsible for enhancing SNePS / CASSIE according to the objectives listed above, using the Blocksworld as an initial development / testing domain. The UMass group was responsible for testing the enhanced system in the specific domain of tutoring.

1.3.1.2 TECHNICAL PROGRESS

Support Software: SNePS, the knowledge representation/reasoning system used for this project, and its associated Generalized Augmented Transition Network (GATN) grammar interpreter/compiler, were implemented in Franz Lisp when this project started. Significant steps had been taken to redesign and reimplement the software, now to be called SNePS-2, in Common Lisp. This was crucial for cooperation between the UB and UMass groups, since both were to use Common Lisp on TI Explorer Lisp Machines. However, when the project started, much work was left to be done.

During the course of the project, the UB group implemented enough of SNePS-2 to be useful (the inference package is still not complete, but is already a superset of Horne Clause Logic), redesigned and reimplemented major portions of it, wrote a usable manual, and, with the help of a related group that the Technical University of Lisbon, included belief revision as a standard feature of SNePS, now called SNePS-2.1.

The UMass group received a copy of SNePS-2.0 early in the course of the project and contributed to its debugging and development. However, mid-way through the project, they ceased to keep up with UB revisions, and maintained their own copy of the system in order to more expeditiously pursue their major research objectives.

Representing Plans and Acts: A representation of plans and associated concepts was designed and implemented. This representation has the following features:

- an act is represented in such a way that a single act can be performed by various actors at various times, and can be part of an actor-independent plan;

- preconditions and effects of acts are represented as properties of the acts – that a particular act has a particular precondition or effect is represented as a simple proposition;
- a goal is represented as a proposition in a particular role, *e.g.* a plan can be asserted to be a plan for achieving a particular goal;
- an act consists of an action and a collection of objects the action is performed on;
- primitive actions are coded into the system, and thus are part of its basic repertoire;
- an act whose action is not primitive is considered to be a complex act – it can only be done by decomposing it into primitive acts (acts whose actions are primitive);
- a plan is a kind of act whose action is one set of control actions, and whose objects are themselves acts;
- that a particular complex act can be decomposed into a particular plan is represented as a proposition.

Actions can be classified as external actions that affect the world, mental actions that affect the system's beliefs, and control actions that affect the system's intentions. Control actions have operational semantics relative to an acting executive. Briefly, the acting executive, which we also implemented, operates on an act queue. Before performing an act, the active executive schedules the achieving of its preconditions and the believing of its effects on the act queue. Control actions are primitive actions that schedule their object acts appropriately. The control actions we implemented are: a sequence of acts; a conditional act; an iterative loop of acts; the act of achieving a goal.

The main domain we used to develop and test our representations was the Blocksworld, whose primitive external actions are picking up a block and putting a block on a support.

Discussing Plans: We modified a previous SNePS/CASSIE grammar so that we could use a fragment of English to instruct the system about Blocksworld, request it to perform acts in that domain, and ask it about plans. For example, the system can understand the following paragraph about the primitive action of picking up a block:

```
   "Picking up is a primitive action. Before picking up a
block the block must be clear. After picking up a block the
block is not clear. If a block is on a support then after
picking up the block the block is not on the support. If a
block is on a support then after picking up the block the
support is clear. After picking up a block the block is
held."
```

Plans for decomposing complex acts can also be explained in English:

```
   "A plan to pile a block on another block on a third block
is to put the third block on the table then then put the
second block on the third block and then put the first block
on the second block."
   "A plan to achieve that a block is on a support is to put
the block on the support."
```

Conditional plans can also be communicated, *e.g.*:

```
   "If a block is on a support then a plan to achieve that the
support is clear is to pick up the block and then put the
block on the table."
```

Not only will the system perform these acts when asked in English, it can answer questions about its plans that amount to a simple kind of advanced or hypothetical planning, *e.g.*:

Person: How would you pile A on C on B?

System: I understand that a plan for performing pile on A and C and B is by performing put on B and a table and then performing put on C and B and then performing put on A and C.

Conditional Plans: A conditional plan, such as

```
   "If a block is on a support then a plan to achieve that the
support is clear is to pick up the block on the table."
```

is represented in SNePS as a rule approximately like the Predicate Calculus rule

```
∀x,y[Block(x) H Support(y) H (On(x,y) ⇒
GoalPlan(Clear(y), Sequence(Pickup(x), Put(x,Table)))].
```

In a situation in which block *A* is on block *B*, and the system must clear *B*, it will derive and store the plan,

```
GoalPlan(Clear(B), Sequence(Pickup(A), Put(A,Table)))],
```

which says that a plan to clear *B* is to pick up *A* and put it on the table.

Since this plan is stored, it would seem that it would be retrieved as a plan for clearing *B* in some later situation when C, for example, is on *B*, and this would be wrong. However, our system is implemented in SNePS-2.1, which includes the assumption-based SNePS Belief Revision System. The plan

```
GoalPlan(Clear(B), Sequence(Pickup(A), Put(A,Table)))],
```

is derived based on the assumptions *Block(A)*, *Support(B)*, and *On(A,B)*. As soon as *A* is picked up, the assumption *On(A,B)* is removed from the current context, and the plan is unavailable until *A* is put back on top of *B*. Thus, the representation of conditional plans is correct in systems that include an appropriate belief revision mechanism.

Plan Recognition: Our representations for plans and acts also facilitate plan recognition. We have implemented a system which allows the deduction of a set of plans some agent might be performing from information about the acts that agent has been performing. This plan recognition system has mainly been applied to a simple version of the Blocksworld, but was also used for the tutoring domain in order to demonstrate how domain knowledge can be used to narrow the number of possible plans some agent might be engaged in.

In order to explore the advantage and disadvantages of using node based inference and path based inference, corresponding rules were implemented and tested in the Blocksworld. The plan recognition system which uses node based inference was able to deal with complex acts and subplans, *i.e.* it can identify a plan even if the reported acts are only implicitly represented in complex acts for which there are separate plan-act rules which contain the reported acts explicitly. The problem with node based inference for plan recognition is that it generates a lot of nodes in order to make the component relations explicit, and it cannot deal with uninstantiated plan-rules due to the current implementation of quantified variables.

Using nested entailments, a left recursive representation for plans, and a special representation for the result of a plan recognition process, it was possible to use mainly path based inference, which increases efficiency and avoids some of the problems related to the quantification of variables in plan -rules. The plan recognition rules were tested successfully for instantiated and uninstantiated plan-rules and for plan-rules with complex acts and corresponding subplans.

Using Plans for Narrative Understanding: The UMass group succeeded in building a system that is able to comprehend and reason about text which involves human interaction, plans, and goals in the domains of tutoring, physics, and international relations. Such texts require more complex representation and more subtle inferencing rules than other simpler texts. For example, the system has to infer the plans and goals that guide human actions and to represent low-level and unspoken actions in paragraphs dealing with tutoring, physics principles, and international relationships. Plan recognition techniques enable the system to understand the intent and nature of the dialogue.

Since this system responds to and reasons about "real life," it must handle uncertainty and dynamic text. It cannot rely upon a pre-organized set of plans or a plan library. Instead, it must perform inferencing, both by using top-down plans and by using domain knowledge. The latter fills in for incomplete plans, plans which involve exceptions, and plans that need to be rewritten, regenerated, or discarded.

In the case of the tutoring text, the system employs deep knowledge of the domain, of discourse strategies, and of tutoring actions. It is a hybrid system, using both traditional plans – small plans representing situation-action pairs – and larger plans to represent hierarchical strategies. It depends greatly upon domain knowledge of tutoring and physics. Strategic plans provide deep expectations of the acts and goals in the text, and reactive planning helps comprehension of more subtle points in the text. The two frameworks supplement and check expectations provided by the other.

The UMass group also developed and implemented a system for resolving Model Anaphora, or reference to sets of objects and actions. Such large activities and text are often referred to by the terms "situation," "the whole thing," or "the example."

1.3.1.4 POSSIBLE EXTENSIONS

So far we have concentrated on the problem of designing representations suitable for discussing, using, and recognizing plans. We have demonstrated their use in a single-agent world. We are now ready to explore issues involved in using our representations to model rational cognitive agents that are capable of acting in the real world. The real world is constantly undergoing change in the presence of several agents (including the modeled one) as well as by natural phenomena. In order to behave as rational agents in a real world they should be endowed with appropriate sensors as well as effectors. Thus we are ready to explore issues concerning sensory acts, external events, and how an agent's beliefs, desires, intentions, and actions are affected by them.

In our current model (and in other state-of-the-art systems) reasoning is performed by some inference engine and acting is done under the control of an acting executive. In order to achieve our goals, we have come to the conclusion that inference and acting need to be more tightly coupled. A survey of most systems will reveal that it is somewhat awkward to do acting in reasoning (or logic-based) systems (but it is convenient to talk about representational and reasoning issues), and it is awkward to research reasoning and representational issues in systems designed for acting and planning. We are beginning to take the viewpoint that logical reasoning rules implicitly specify the act of believing, and the process of reasoning can be treated as specialized (more efficient) acting. This will enable us to integrate the acting and inference engines that can be driven by regular reasoning rules as well as connectives that will transduce a belief status to an intention-to-act status. We are currently designing such connectives. Thus our future research will attempt to clarify the relationship between inference and acting. This integrated approach used in conjunction with the principles underlying propositional semantic networks will preserve the power of acting, as well as reasoning systems, and provide a richer framework within which one can experiment with various modeling issues in AI.

A second direction for our future research involves a reexamination of the representation of variables in SNePS. Consider again the conditional plan,

```
∀x,y[Block(x) H Support(y) H (On(x,y) ⇒
    GoalPlan(Clear(y), Sequence(Pickup(x), Put(x,Table)))].
```

As in the FOPC representation of this rule, the SNePS representation contains the subexpression *Pickup(x)*. Although in SNePS, the variable x is connected in the network to its restriction, *Block(x)*, the variable x is still an "atomic" node, and the term *Pickup(x)* does not contain the restriction on x as a subterm of it. The significance of this is that the act "pick up a block" is not represented by a single term in the plan expression. Compare this representation to something like:

```
GoalPlan(Clear(y:Support),
        Sequence(Pickup(x:Block s.t. On(x, y:Support)),
                Put(x:Block s.t. On(x, y:Support), Table)))
```

Here, each sub-expression is conceptually complete. For example, `Pickup(x:Block s.t. On(x, y:Support))` clearly represents the act of picking up a block that is on a support. (It should be noted that the SNePS representation, using a network syntax, would not be as redundant as the linear representation). We plan to investigate these representational issues further.

A more application-oriented direction we may pursue is to apply our techniques of representing and reasoning about plans to simulate some human agent, and to try to predict what that human agent would do in certain hypothetical situations.

1.3.2. Versatile Maintenance Expert System (SUNY Buffalo): Vol. 3

The State University of New York at Buffalo started its participation in the Northeast Artificial Intelligence Consortium in 1984 with the objective of developing a versatile expert system for equipment maintenance. A prototype expert system was designed to advise a maintenance technician on testing. However, during the course of this project, it evolved into a more versatile system by incorporating features such as model-based reasoning and communication capabilities such as natural language and graphics. The new system came to be known as the Versatile Maintenance Expert System (VMES).

VMES research is concerned with the development of a system that could diagnose faults in an electronic circuit and interact with a maintenance technician.

Versatility has been the main goal of our research. VMES is designed to be versatile across a range of maintenance levels and across a variety of user interfaces. To achieve these versatilities, the device model-based approach has been followed. VMES has been implemented in SNePS, the Semantic Network Processing System, and has several modules: an expandable component library as its knowledge base, an inference package with diagnostic rules, an active database for diagnosis, a user interface for intermediate users to adapt VMES to new devices by incrementally updating the component library, and a multimedia user interface for end users with VMES for fault diagnosis.

Our accomplishments during the lifetime of the project (1984-1989) can be classified into the following seven categories:

1) Device modeling – structural and functional knowledge and efficient representation;

2) Graphical interface for end users to interact with VMES;

3) Model-based reasoning for diagnosis – initial candidate ordering, reordering, and elimination;

4) Sequential circuit representation and a general control structure for diagnosis;

5) Representation and diagnosis of a real device;

6) Migration of deep knowledge to shallow knowledge; and

7) Enhancement of SNePS, the system used for the implementation of VMES.

In the remainder of this summary, a few more details in each category are given.

Device Modeling: VMES uses structural and functional descriptions of devices to avoid difficulties of empirical rule-based diagnosis systems in knowledge acquisition, diagnosis capability, and system generalization. Based on the requirements that the system be expressible, buildable, usable on a computer, and expandable, a device in the circuit domain has been modeled as a hierarchically arranged set of subparts from both logical and physical perspectives. Wires and points of contact (POCONs) have been explicitly represented in order to perform the diagnosis of faults in circuit connections.

Graphical Interfaces: The usability of VMES would be enhanced greatly if it were to communicate with the maintenance technician graphically as well as with text. Toward that end, we developed a general theory of "Graphical Deep Knowledge," defined as declarative knowledge that is projectively adequate (adequate for drawing) as well as deductively adequate (adequate for deducing relevant spatial information). We also made major progress in the field of Natural Language Graphics, the attempt to design a knowledge representation that may be used for graphical deep knowledge and for meaning structures that underlie the comprehension and generation of natural language. Using these theories, we designed and implemented graphical interfaces for VMES that can draw the devices being tested, and that can graphically show the reasoning process that VMES is pursuing to diagnose the device.

Model-based Reasoning: A major step in model-based fault diagnosis has been the generation of candidate submodules which might be responsible for the observed symptom of malfunction. After the candidates are determined, each submodule can then be examined in turn. It is useful to be able to choose the most likely candidate to focus on first so that the faulty parts can be located sooner. We have developed a systematic method for candidate ordering that takes into account the structure of the device and the discrepancy in outputs between the observed and expected values. However, because the same good/bad output pattern of a device always gives rise to the same initial ordering, the method has its limitation. For any device and good/bad output pattern, it is easy to come up with an example on which the method does poorly in the sense that the actual faulty part is in the last place of the initial ordering.

To fix this problem, more dynamic methods, candidate ordering and elimination, have been developed. Both methods modify the candidate list as new information becomes available. Reordering moves components connected to the consistent inputs of the current candidate to the front of the candidate list and those connected to the inconsistent inputs to the tail. Elimination removes components which no longer have any non-error-propagating paths to incorrect primary inputs after the current candidates is found not to be error-propagating. It has been proved that under the single fault assumption, the average number of components to be checked is $O(\log n)$, where n is the number of components in the circuit under test. In general, $i \log n$ components will be checked when there are i faults in the circuit. As to the implementation, all the above-mentioned theories have been implemented in our VMES system. The system has been successfully ported from our Unix machine using SNePS-79 and Franz Lisp to TI Explorers using SNePS-2 and Common Lisp. The improvement in performance is enormous: the system runs at least ten times faster than before.

An interface for encoding devices represented using structural templates and instantiation rules has been implemented to facilitate fault diagnosis. This is user-friendly and robust, providing for as few key strokes from the user as possible. It fills in most of the invariant template, documents the code, and stores it in the appropriate file for the user. It was encoded in Franz Lisp on a VAX to begin with, and was transferred to Common Lisp on a TI Explorer.

Sequential Circuit Diagnosis: In order to incorporate sequential circuit diagnosis into VMES, the following steps have been taken: (1) change in device knowledge representation; (2) change in control structure and inclusion of assumption relaxation; and (3) candidate generation based on electrical behavior. The change in representation was essential for better organization of the device knowledge and the incorporation of sequential components. Devices no longer have any physical hierarchy, and logical hierarchy is not related to physical details of the device. This has enabled an arbitrary number of logical levels, and has allowed arbitrary grain of focus for diagnosis. Wires were no longer represented explicitly. Diagnosis of wires and POCONs was intended to be hard-coded into VMES. Assignable variables were found to be necessary to represent states in sequential components. Since SNePS is built on the philosophy of logical programming, it did not allow for such variables. Hence, the new scheme of representation had been developed in Common Lisp.

The control structure has been streamlined so as to accept various options such as shallow reasoning and diagnosis with multiple symptoms. Explanation is a necessary part of an expert system and so an explanation generation was also incorporated. The criteria upon which the system should discard single-fault and non-canceling fault assumptions has been worked out and partially implemented. The current candidates is found not to be error-propagating. It has been proved that under the single fault assumption, the average number of components to be checked is $O(\log n)$, where n is the number of components in the circuit under test. In general, $i \log n$ components will be checked when there are i faults in the circuit. As to the implementation, all the above-mentioned theories have been implemented in our VMES system. The system has been successfully ported from our Unix machine using SNePS-79 and Franz Lisp to TI Explorers using SNePS-2 and Common Lisp. The improvement in performance is enormous: the system runs at least ten times faster than before.

An interface for encoding devices represented using structural templates and instantiation rules has been implemented to facilitate fault diagnosis. This is user-friendly and robust, providing for as few key strokes from the user as possible. It fills in most of the invariant template, documents the code, and stores it in the appropriate file for the user. It was encoded in Franz Lisp on a VAX to begin with, and was transferred to Common Lisp on a TI Explorer.

Sequential Circuit Diagnosis: In order to incorporate sequential circuit diagnosis into VMES, the following steps have been taken: (1) change in device knowledge representation; (2) change in control structure and inclusion of assumption relaxation; and (3) candidate generation based on electrical behavior. The change in representation was essential for better organization of the device knowledge and the incorporation of sequential components. Devices no longer have any physical hierarchy, and logical hierarchy is not related to physical details of the device. This has enabled an arbitrary number of logical levels, and has allowed arbitrary grain of focus for diagnosis. Wires were no longer represented explicitly. Diagnosis of wires and POCONs was intended to be hard-coded into VMES. Assignable variables were found to be necessary to represent states in sequential components. Since SNePS is built on the philosophy of logical programming, it did not allow for such variables. Hence, the new scheme of representation had been developed in Common Lisp.

The control structure has been streamlined so as to accept various options such as shallow reasoning and diagnosis with multiple symptoms. Explanation is a necessary part of an expert system and so an explanation generation was also incorporated. The criteria upon which the system should discard single-fault and non-canceling fault assumptions has been worked out and partially implemented. The new system VMES II, with the above modifications, runs considerably faster than the previous system.

Since sequential circuit diagnosis is harder than combinational, the stage of candidate generation had to be exploited to the maximum so as to narrow the focus of diagnosis as much as possible. Therefore, candidate generation based on electrical behavior was researched and implemented. In this method, given the symptom, the system works backward through the subdevices to come up with a probable candidate list. The scheme is based on the assumption that one could shorten the suspect list by eliminating those of the relevant inputs that can in no way contribute to the observed system at output. This knowledge can be conveniently expressed for small devices as fault characteristics and was exploited during diagnosis of large systems.

Sequential circuit diagnosis poses many special problems. A sequential circuit has feedback loops, and therefore it is necessary to specify the start and end of the loops for proper simulation of states. Moreover, during simulation and candidate generation, more than one (as many as the clock cycles) value has to be stored with each device port. Fault characteristics are not as well behaved for sequential circuits and are to be specified for the superdevice as well. These and many other related problems have/are being solved towards incorporating sequential circuits.

Diagnosis of Printer Buffer Board: Although VMES was intended to be adaptable to a wide range of devices in the domain of digital circuits, selection of a realistic device was as a test bed. Upon the recommendation of RADC, a Heathkit Printer Buffer Board was selected. This test device, assembled locally, is of reasonable complexity to work with. It consists of an eight bit microprocessor, two sets of serial and parallel ports, memory, and latches. This device has been represented hierarchically at various levels of abstraction, and diagnosis has been carried out on parts of the circuit. A device of this kind helped spur new ideas, extensions, and refinements to the diagnosis theory.

Migration of Deep Knowledge to Shallow Knowledge: Deductive reasoning systems, including automatic fault diagnosis, usually make use of vast numbers of rules to infer new knowledge from existing knowledge. Different rules may be at different levels of generality. In particular, a method of rule nesting enables a rule to contain a subrule in its consequent. If some instance of the antecedent of the outer rule is satisfied, that instance of the subrule will be asserted in the knowledge base. This subrule has a lower level of generality than its dominating rule. We call this phenomenon the migration of deep to shallow knowledge.

As more specific knowledge emerges from the general knowledge, the process of inference might become slower if the system tries to use both specific and general rules for similar problems. We need a proper scheme to take advantage of specific knowledge to avoid duplicate inference branches and, in the long run, to realize fast inference. An idea of shadowing general knowledge has been suggested and implemented by exploiting its instance information accumulated in the previous reasoning. A list of instances containing the binding information was maintained for each rule so that this rule can be blocked in the next inference if instances of that rule are already activated with the same binding information. This shadowing strategy is tested for wire faulty detection rules of the M3A2 circuit, and the result shows a significant improvement in diagnosis speed for the same types of wires.

Enhancements of SNePS: During the course of this project, we have continued the development of the Semantic Network Processing System. This effort resulted in a new major version of the system called SNePS-2.0, and a new minor version, called SNePS-2.1. SNePS-2 represents a major redesign of SNePS, and is implemented in Common Lisp for maximal portability. SNePS-2.12 incorporates an assumption-based belief revision system as a standard feature of the system. We have ported SNePS to, and maintained it on, over seven different computer systems.

Possible Extensions: The current VMES system can be enhanced in a number of ways. The control structure can be generalized to include various schemes of diagnosis. For instance, a retry method, or direct isolation or intersection isolation can be applied in a sequence to diagnose a circuit. This involves the development of direct and/or intersection isolation techniques and its implementation. Model-based reasoning can be enhanced by incorporating heuristic knowledge in the reasoning process. It is also worthwhile to investigate the effect of multiple tests, test generation techniques, and mixing of procedural and declarative knowledge in diagnosis. The versatility of VMES can be further enhanced by the development and integration of analog circuit component diagnosis. Diagnosis of intermittent faults is another area that requires further investigation.

1.3.3 Distributed AI for Communication Systems Control (Clarkson): Vol. 4

This volume describes the results of a five year research project performed at Clarkson University as part of the Northeast Artificial Intelligence Consortium (NAIC). The major objectives of this research were to gain a better understanding of issues that arise in the area of distributed artificial intelligence (DAI) and to investigate the application of DAI in communications network management and control. This report gives a brief overview of the problem domain, describes the development of a DAI testbed (DAISY), and presents results which have made contributions to DAI and to distributed network management. The report also documents the building of an AI infrastructure which will support the growth of future AI research at Clarkson, both in conjunction with other universities in the NAIC and with the U.S. Air Force at Rome Air Development Center.

Distributed artificial intelligence is concerned with issues that arise when loosely coupled problem solving agents work collectively to solve a problem. These agents are typically characterized as having a degree of functional specialization, a local perspective, and incomplete knowledge. Although each agent uses its own local perspective and knowledge in performing its tasks, a complete solution to the problem usually requires cooperation among the agents. One of the most important issues in these problems is which cooperation paradigms are most appropriate.

The strategy followed in this research was first to choose an application problem domain to which DAI might be applicable. This domain should have a structure and degree of complexity that make it a realistic one for investigating these problems. The monitoring and control of large communications systems was selected because it is a naturally distributed, complex problem which currently requires cooperation among humans for effective solution. The problem is an important one to the Department of Defense, and it is rich both in problems to be solved and in structure that can be used to investigate distributed problem solving paradigms.

The model we have used for a communications network is based on the large scale, world-wide Defense Communication System (DCS). We have concentrated on network management and control at the subregion level. The subregion level represents a group of several individual sites or nodes in the communications system architecture which are monitored and controlled from a single control center. System-wide management and control is distributed over a network of subregion control centers, typically from three to twelve in number. Our view of the role of DAI in this environment is to provide cooperating, intelligent, semi-autonomous agents to serve as problem solving assistants to the human controllers.

We have developed an architecture for this system in which each component (an agent) is a specialized and localized knowledge-based system designed to provide assistance to the human operator. Analysis of the problem solving activities currently employed by these operators found three fundamental kinds of functions required: (1) data interpretation and situation assessment; (2) diagnosis and fault isolation; and (3) planning to find and allocate scarce resources for restoral of service in the event of an outage. This suggested a natural functional distribution of agents. Not only must each agent be able to cooperate with other agents performing different functions at the same local site, but each agent must also be able to cooperate with identical agents located in physically separate facilities. This concept of a knowledge-based network management system as a spatially and functionally distributed collection of semi-autonomous, cooperating intelligent agents is an important contribution to the development of future network management and control systems for the DCS.

At a local level, the system is seen as a number of functionally specialized agents that cooperate in a loosely coupled fashion. These agents comprise a local participant in a network-wide team of problem solvers. At the global level, the system may be viewed as a group of relatively independent, spatially distributed problem solving systems cooperating to solve a collection of problems.

An important feature of the system is cooperation of agents. Two general methods for cooperation have been explored. First, problem solvers may cooperate through the exchange of messages. Agents may coordinate their actions by requesting another agent to perform some task in order to achieve a goal. Cooperation of agents may also take the form of an exchange of knowledge through messages when one agent needs additional knowledge in order to further problem solving activity. The second mechanism for cooperation is through sharing local knowledge about the current state of the network and the status of problem solving activity. Inferences of one agent are shared with the others, in a central knowledge base. The shared knowledge base is managed by a knowledge base manager.

In order to implement this architecture in a laboratory environment, we developed a Distributed AI System (DAISY) testbed which supports simulation of multiple agents on one or more LISP processors. The DAISY testbed incorporates two system building tools which we developed during this effort. SIMULACT is a generic tool for simulating multiple actors in a distributed AI system. It is domain independent, permits rapid prototyping of distributed software modules, and allows interactive experimentation incorporating a gauge facility for monitoring and data collection. A graphical user interface (GUS) was developed to assist a user in capturing the structural knowledge about a communications system. A typical communications network consists of thousands of objects which are interrelated by the structural organization of the network. GUS enables a communications network expert to describe the details of this organization using a familiar set of graphical symbols.

Our primary results have been in the development of cooperation paradigms for distributed agents. As an example of cooperation through message exchanges, the service restoral task requires agents which perform distributed planning subject to constraints imposed by network topology and resource availability. We have developed a distributed multi-agent planner (DMAP) which extends the current work in planning by designating certain objects as resources so that they may be efficiently allocated for effective use in satisfaction of multiple goals. The planner consists of two stages, plan generation and multistage negotiation. During plan generation, agents are required to generate plans which utilize limited system resources in a domain where both the knowledge about resources and the control over these resources are distributed among the agents. Experimental results are presented which show that plan generation in this class of problems can be accomplished by the exchange of a limited amount of information between agents. It is unnecessary for any single agent to acquire complete global information about the system.

After a set of plans has been established, agents must cooperatively select specific plans to execute as many goals as possible, subject to resource constraints. Multistage negotiation has been developed as a means by which an agent can acquire enough additional knowledge to reason about the impact of local decisions on nonlocal system state and thus modify its behavior accordingly. Because no single agent is in control and no single agent has complete knowledge of the entire system state, an important aspect of multistage negotiation is the mechanism for providing agents with nonlocal information. We have developed a formalism for abstracting and propagating information about the nonlocal impact of decisions made locally. Our work provides mechanisms for determining impact at three levels: locally on the level of plan fragments, locally on the level of goals, and nonlocally. This approach may be viewed as promoting cooperation among agents by using constraint-based reasoning to develop good, local heuristic decision making.

A second example of our results in the study of cooperation methods involves sharing knowledge among local agents. We have implemented a method by which knowledge can be shared in a local knowledge base in the form of inferences and default assumptions. Specifically, a Multiagent Assumption-based Truth Maintenance System (MATMS) has been developed to manage a knowledge base shared by multiple problem solvers. The most important feature of the MATMS is that it provides the foundation for resolving inconsistency between agents, while supporting the notion that two problem solvers can have different views concerning the state of a particular piece of knowledge. The MATMS handles differing views by allowing independent belief sets for each of the agents. It supports resolving inconsistency between agents by providing a mechanism for comparing two agents' belief sets. An agent's belief set is characterized as the default knowledge base (which is common to all agents) with an overlay placed upon it. The MATMS is efficient largely because it focuses its efforts on managing these overlays, not the entire belief set of an agent. By concerning itself only with the overlays, the MATMS can switch from addressing one problem solver's belief set to addressing another's expeditiously. It can also change an individual problem solver's belief set quickly, because the default knowledge is not explicitly carried over from one belief set to another.

The third area of significant results for distributed cooperation is the development of a distributed theorem prover. This represents some of our most recent work and is thus still in an early stage of development. We have implemented a distributed theorem prover as a model for cooperation among agents performing a distributed situation assessment task. Each agent has some limited view of the network state and thus may form an hypothesis of the global network state. However, in order to confirm this view, additional knowledge obtained from other agents is necessary. Confirmation of this view becomes a theorem to be proved. Our preliminary results indicate this approach appears to be very promising.

This report also presents results of interest to the future development of distributed network management and control systems. In addition to the architectural considerations mentioned above, our work has shown that an object-oriented database in combination with problem solving knowledge is an appropriate representation tool for knowledge-based systems in the area of network management. We have also demonstrated the value of an interactive, graphical tool, such as GUS, for assisting the knowledge engineer in capturing the many levels of detailed knowledge necessary in completely representing communications network knowledge. By close cooperation with other contractors working on development of actual prototype systems, we have assisted in the transfer of this technology to the commercial and military technology base.

The building of an infrastructure to support the continued growth of AI research activities among the NAIC universities was an important ancillary objective of this research project. At Clarkson we have made significant developments in terms of the growth of faculty activity in AI research, offering of new AI courses at both undergraduate and graduate levels, addition of new research facilities, and an increase in the numbers of graduate students working in AI. During the past five years we have formed new working relationships with research groups at other NAIC universities and with other contractors to RADC. These relationships have strengthened our abilities to conduct meaningful research and to assist the transfer of technology from the university research lab to the industrial development environment.

Finally, this report looks ahead to future research problems. While we believe we have made significant progress during the past five years, the work started then is not yet complete. As a result of what we have learned about knowledge representation and acquisition, we have new ideas for the design of a local knowledge base for network management. The importance of graphical tools was underestimated in our original design of the DAISY testbed. Since the human operator or user is expected to continue to play an important role in these systems, an improved interface between intelligent agents and the human is critical to the development of a successful system. With the basic testbed now defined, and component parts developed, there remains much work to be done in testing the initial design choices under a variety of adverse operating environments. We have yet to determine the robustness of our cooperation paradigms under stress. Thus this report should be looked upon not as the conclusion of this work, but rather as having set the stage for new work in investigating the design of distributed AI systems. These systems have application not only in communications network management, but also in other areas of command, control, communication, and intelligence information processing.

1.3.4 Distributed Planning for Dynamic Environments in the Presence of Time Constraints (Clarkson/UMass): Vol. 5

This task is one which started in August of 1988. Our primary goal has been one of developing a testbed environment that is appropriate for research in real time distributed planning. The problem domain selected as an example context in which to investigate the issues of real time distributed planning is forest firefighting. Consequently, much of our activity has been devoted to development of a multiagent firefighting simulation. This was necessary in order to provide an environment in which agents can cooperatively plan to contain fires. In order to accomplish this task, a number of issues related to timing, agent synchronization, management of "thinking time" and "acting time", and agent capabilities have been addressed.

The testbed simulator has not been designed simply as a distribution of an existing centralized firefighting simulator. We found that the issues of time and agent synchronization in a multi-agent environment necessitated a complete redesign of the existing firefighting simulator. The new design permits multiple agents to work simultaneously and independently. It includes a facility for defining the characteristics of communications among agents, with available communication media independently specified. In addition, the new simulator reflects a much more realistic terrain representation and a significantly improved fire model than the previous centralized simulator.

The research efforts on this task have been concentrated on formulation and implementation of an appropriate agent model and mechanisms for handling time in general and reasoning strategies for adjudicating allocation of time among various cognitive activities in the planner. It seems clear that time is a critical resource for these types of problems. When time is viewed as a resource, proper allocation of time among subtasks is critical in achieving reasonable performance. One problem central to development of heuristics for determining time allocations is that of formulating ways of handling the fact that time can be viewed in more than one way. It seems evident that the CPU time associated with the planner is measured in seconds to minutes, whereas the time associated with acquiring information regarding the state of the fire may be measured in minutes or hours. Reasoning about actions in an environment such as this requires that the planner understand and be able to deal with these extreme differences in scale. Effectively, there are two types of time: "execution time" and "action time" or "internal time" and "external time." These two types of time share some attributes, but are fundamentally different in others (as far as the planner is concerned). Research concerning a model of distributed planning, agent characteristics, and effective ways of modeling time was initiated and mechanisms for experimenting with time (as perceived by agents in the system) have been incorporated in the simulator design. In addition, qualitative reasoning about time allocation has been investigated and appropriate algorithms have been implemented. Finally, an investigation of ways in which multiple "firebosses" can be accommodated has been initiated.

1.3.5 Building an Intelligent Assistant: The Acquisition, Integration, and Maintenance of Complex Distributed Tasks (UMass): Vol. 6

Our research over the five year period has focused on two basic and interrelated research questions: 1) how to automate the acquisition, integration, and maintenance of a global understanding of a complex process from multiple, distributed local perspectives, 2) how to use this "understanding" to support users in their cooperative interaction with the system and other users by assisting in the execution of tasks and by explaining the reasons behind the actions and decisions involved in reaching the current state of the system. We believe these are some of the key research questions that need to be solved in order to effectively use a distributed network of workstations to perform cooperative problem solving in a complex task-oriented environment. The importance of this problem area and related research issues has been increasingly recognized over the five years of this research project. We think that the results of our research will, and to some degree already have had, a significant impact on the emerging subfields of intelligent interfaces and computer-supported cooperative work.

During the five years of the project, we have studied these issues by looking at the task domains of software development, office procedures, project management, and tutoring. We have built a number of systems to demonstrate our research. Our early work in an intelligent assistant focused on understanding user actions through hierarchical event-based descriptions. An offshoot of this work was the approach of using event-based behavioral abstractions as a tool for distributed debugging and performance evaluation. This approach is now receiving wide attention. As a result of this early work, we realized that advances in the following basic research areas would be necessary for satisfying the goals of our project. For each of these areas, we list our research accomplishments:

KNOWLEDGE REPRESENTATION – The goal is to provide a framework for representing realistic models of complex open-ended domains. The results were:

- a knowledge representation framework that integrates activity models, agent descriptions, object specifications and relationships;

- meta-plans as a specification technique for large and complex plan libraries and domain-dependent exception handling routines;

- integration of empirical knowledge which represents soft domain constraints into classic hierarchical plan formalisms through the addition of a Truth Maintenance System (TMS);

- integration of planning and simulation techniques for validating plans in complex, dynamic environments.

KNOWLEDGE ACQUISITION – The goal is to develop techniques for user specification of plans and dynamic acquisition at run time of new and revised plans. The results were:

- an approach for knowledge acquisition based on the assimilation of new and revised plans with existing specifications;

- a cognitive model for how people recall their activities and an interface based on this model that can be used to acquire plans, display plans and modify plans;

- a formal model of plan exceptions, and techniques for detecting, classifying and learning from them.

FOCUSING IN PLAN RECOGNITION – The goal is to develop techniques for quickly and efficiently arriving at the best interpretation of the actions/data that have currently been observed. The results were:

- a recognition architecture that exploits heuristics based on user rationality and fully uses available constraints as soon as possible in the recognition process;

- a new approach to controlling plan recognition, called evidence-based plan recognition, that uses a symbolic representation of current uncertainties in the interpretation and control plans keyed to specific uncertainties to efficiently guide the recognition process.

MULTI-AGENT, INTERACTIVE PLANNING – The goal is to provide a framework to specify partial plans and for the user(s) to interact with planner(s) to complete these plans based on the dynamics of the specific situation. The results were:

- an architecture based on a formal model of interactive planning has been implemented;
- models for negotiation among user and system, and among systems, have been developed.

KNOWLEDGE DISPLAY – The goal is to provide a display framework and tools for the user to effectively interact with an intelligent assistant:

- a suite of programming tools that enables authors to browse and explain knowledge in an expert system for tutoring. These tools facilitate tracing and summarizing the reasoning within an expert system and allow an author to interactively modify system reasoning and response in an intelligent discourse system;
- a graphics object-oriented environment for building simulations of complex environments for decision support.

These ideas have been realized in a number of systems. The GRAPPLE system monitors a user's activities, detects errors, and reasons about the user's plans. It uses domain knowledge to make plausible assumptions about missing values in an open world application. This is used to provide improved error detection, prediction, and disambiguation.

POLYMER is a planning system which constructs partial plans and executes them interactively. It uses constraints from agent actions to extend its partial plans. Exception handling is achieved by a subsystem called SPANDEX, that classifies exceptions and constructs an explanation of how each action may fit into the current plan. Plan acquisition is supported by a subsystem called DACRON, that has a graphical interface based on how people recall their tasks, and a knowledge assimilation is supported by a subsystem called KnAc.

In short, we feel that significant progress, both from a conceptual and practical perspective, has been made in developing an intelligent assistant to support knowledge-based, computer-supported cooperative systems. Additionally, we think there has been important spin-off research in planning and plan recognition that will also have significant impact.

1.3.6 Automatic Photointerpretation (RPI): Vol. 7

Matching of models to image features is a fundamental step in computer vision systems. Such matching may take place a different levels of these systems, from template matching of raw images to symbolic matching of relational models. In this volume, the problem of matching localized spatial features with arbitrary attributes sets to either idealized or learned models is addressed. The minimum representation criterion used to achieve an acceptable match is a principal topic of this report.

Image matching is difficult to achieve with sufficient generality, speed, and robustness to be useful in practical systems. Many proposed algorithms are highly dependent on a choice of particular features and model representation, and they often require interactive or heuristic methods to extract features. Adding generality to matching procedures has been difficult particularly because evaluation function or match quality measures do not generalize well. Image matching is inherently complex from a computational point of view, since the number of possible matches in general grows exponentially with the number of features. Polynomial complexity is an important property of any practical approach.

Good image matching algorithms must be able to handle feature uncertainty including missing data, extra features, and noisy attributes. This requirement has been particularly difficult to achieve since most evaluation functions are not able to handle missing or extra data in a consistent non-heuristic fashion. The representation criterion presented in this report is inherently normalized to match size and number of attributes and directly accommodates missing and extra data.

This report describes the minimum representation criterion as a basis for image transformation and correspondence matching. We specifically address the problem of two-dimensional rigid, unattributed point sets with missing and extra points. The algorithms developed are polynomial in complexity and near-optimal for this criterion. Examples of performance on highly variable gray-level images including aerial imagery are shown. Results which have been obtained on the application of minimum representation matching techniques to several types of imagery including aerial photographs obtained from RADC are summarized. While the underlying methodology for the minimum representation approach has been developed previously, the current work has emphasized a new implementation of the work and application to new types of imagery. This report includes an overview of the basic methodology, new implementation, and new applications, and augments the papers which have been prepared summarizing our results.

In Volume 7, section 7.2 defines the image matching problem. Section 7.3 presents the minimum representation criterion principles. Section 7.4 describes a usually optimal, polynomial time algorithm for image matching and transformation. Section 7.5 presents some examples of the matching procedure.

1.3.7 Speech Understanding Systems (RIT): Vol. 8

Rochester Institute of Technology's contribution to the NAIC has been the research and development of a system of techniques and processes suitable for use in a continuous speech, large vocabulary, speech understanding system. We have incorporated these techniques into a functioning workstation which is capable of testing, evaluating, and delivering this speech understanding technology. Other contributions made possible by this NAIC project have been: the engendering of AI capabilities within Rome Air Development Center (RADC), the support and educational growth of many students and researchers at RIT who worked on the project over the past five years; and private industry involvement.

The research and development of RIT's speech understanding system carefully incorporated testing and evaluation methods at each level of planning, design, implementation, and testing. These methods allowed us to not only produce the optimal integration of these technologies, but also produced qualitative and quantitative comparisons of less successful techniques so that future researchers might benefit from our extensive testing. This comparative work was performed at all levels of system development including the system architecture, control structure, knowledge representation, implementation, and error analysis.

This comparative evaluation methodology required us to design a highly modular framework in which we could prototype and evaluate the speech understanding techniques that were being developed. A hierarchical system with multi-level knowledge representations was chosen as the best approach for handling this type of comparative development. The interfaces at each level were derived from the knowledge representation of the speech used at that level of hierarchy. The knowledge representations at each level were derived from the levels of data reduction that occurred as the speech was processed along the continuum of raw acoustic waveform to a representation of meaning. The well-defined interfaces and modular programming approach allowed head-to-head comparisons of several techniques within each level of the system hierarchy.

The system development was made possible by the use of the ESPRIT (Explorer Speech Processing at RIT) system. ESPRIT is a speech research development environment which runs on the Texas Instruments Explorer workstations. ESPRIT was developed at RIT, for use as a test-bed for the speech understanding system. The ESPRIT environment provides researchers unfamiliar with Lisp, and the Explorer workstations, the ability to develop speech and signal processing experiments. ESPRIT uses a mouse-and-menu interface combined with a graphical programming language to both design and operate a variety of speech and signal processing experiments. The work on the ESPRIT system has also led to the development of an object-oriented simulation workstation at RIT Research Corporation and has involved private industry (Texas Instruments and Allied Signal).

Our research has involved researchers from many disciplines. The fields of artificial intelligence, electrical engineering, speech audiology and phonology, mathematics, and statistics were all represented in some aspect of the project. Some of the methodologies that have been studied are expert systems, neural networks, hidden Markov models, conceptual analysis, dynamic programming, and statistical classification techniques.

Our prototype speech understanding system is a knowledge based system which attempts to capture the knowledge the experts use in reading and interpreting spectograms. This knowledge allows us to generate phonetic information from the raw acoustic waveform. From the phonetic transcriptions, words are hypothesized and these utterances are used by the natural language system. The natural language system then analyzes the utterances and produces a representation of meaning for the utterance.

The system has been designed to be domain-independent. We have found it necessary to introduce domain-specific information at the higher level understanding functions, but this is not unusual in natural language understanding systems. The lower levels of the system hierarchy were, in fact, tested using a completely different domain than that used with the higher level understanding functions. We feel that our domain-independent approach makes the architecture of our system flexible as well as extensible.

The NAIC funded project has produced the following items:

- A prototype system with functionality and competing methods at each level of the system hierarchy.
- Four completed Master of Science degrees, with four more pending.
- The ESPRIT speech processing system. ESPRIT evolved as a byproduct, as it was not funded directly by the NAIC, but its development was necessitated by the NAIC project. The funding was provided by RADC and Texas Instruments, Inc.
- Technology transfer from RIT to RADC as well as to private industry.

We feel that our work has effectively investigated the types of extremely difficult problems encountered when dealing with a large vocabulary, continuous speech, speaker-independent system. There are, however, some areas of research where we feel that further investigation might yield interesting results. These extensions to the work might include: the development of a commercial quality speech understanding system based upon our prototype system, the incorporation of adaptive processes and learning into the system, and the testing and evaluation of as yet undiscovered speech understanding procedures.

Our work in the speech understanding area has allowed us to develop tools, technologies, and personnel that may be applied to other speech related disciplines. The speech understanding work we have done has extensions in the areas of speaker identification, language identification, and key word spotting.

1.3.8 Planner System for the Application of Indications and Warning (Colgate): Vol. 9

The Colgate University project in the framework of the Northeast Artificial Intelligence Consortium (NAIC) was devoted to the design of a planner system for the application of I&W (Indications and Warning). The specification of the task evolved from the early direction of intelligent database management toward the emphasis on problem-solving activity. The task of the project was two-pronged:

A. Design of a system to

1. obtain as input messages concerning events in a model of a real-life subworld;

2. 'understand' these events by detecting what *plans* they are parts of and, whenever applicable, what *goals* are pursued by the instigators of these events;

3. produce (suggestions for possible) plans of action necessary in connection with the situation in the world.

B. Implementation of this system for the world of I&W.

This general task included a large number of subtasks, many of which require significant research effort. We concentrated on designing the mechanisms and knowledge bases for the problems of plan recognition (a part of 'understanding' in 2. above) and plan production (in the framework of 3. above). We excluded from our consideration the problems of perception (speech, graphical, or visual); the problem of understanding natural language inputs (that is, understanding the contents of these messages), as well as actual performance of plans suggested by our system.

The conceptual background of this effort is described in some detail in Section 4. The historical background of the project and the way in which it merges with other efforts in the Consortium is briefly discussed here. Cooperation with other research teams within the Consortium, especially with the University of Massachusetts project led by Victor Lesser and Bruce Croft, led to the state of affairs where plan understanding and plan production has become the main thrust of the research effort at Colgate. It was decided, in consultations with the project monitors at RADC, that the natural language aspects of the task would be postponed for later consideration.

STRATEGY

We have taken a concentric approach to the task of designing and implementing the planning system. In other words, we decided to produce an implementation for every design version of our system (called, for historical reasons, POPLAR). As our study of the problem of knowledge-based automatic planning progresses, newer versions of the system will appear.

In Volume 9 we describe in succession the design peculiarities and the implementation characteristics of the two versions of our system (POPLAR 1.3 and POPLAR 2.0) that were developed. Goals were set for the implementation of the next version of the system (POPLAR 3.0). This last version would have introduced very substantial changes to the overall design, and would have been addressed in the final year of the project if it had continued.

1.3.9 Time-oriented Problem Solving (UofR): Vol. 10

The unifying theme underlying all the research carried out under this project is that of producing new knowledge representation formalisms that extend the range of situations in which problem solving systems can be applied. These new formalisms have centered around issues related to time, temporal reasoning, and causation.

Significant progress has been made in the following areas:

- The axiomatic specification of an interval-based theory of time that allows for two different forms of time points;
- The development and public release of TIMELOGIC, an implementation in Common Lisp of the interval logic that quickly computes the relationship between arbitrary intervals using a constraint propogation algorithm;
- The development of a non-reified first-order temporal logic that has a well defined syntax, semantics, and proof-theory, and is easily implemented using a type-based theorem prover such as RHET;
- The development of a logic that can represent and support reasoning about simultaneous interacting actions;
- A generalized model of plan recognition, both as a formal theory and as a family of practical recognition algorithms;
- The formal specification of two distinct forms of abstraction for planning systems, one based on reduced models, and the other an extension of inheritance hierarchies;

- A new approach to causal reasoning that rejects overly-strong domain-independent approaches to solving the frame problem in favor of domain-dependent "cause-closure" axioms;

- A statistical/probabilistic approach to action reasoning that explicitly models that action success is not guaranteed, but subject to failure some proportion of the time;

- The development and initial implementation of a problem solver that reasons, acts, and senses in real-time within the ARMTRAK model train domain; and

- The development and public release of the HORNE representation system and its extension, RHET, two hybrid logic/frame-based representations for use in general problem solving and natural language understanding.

This work has been well received by the international artificial intelligence community, as evidenced by the publication of over two dozen articles and chapters in international journals and conferences. In addition, in recognition of their outstanding contributions to the field, two of the researchers on this grant received some of the highest honors that can be given to computer scientists: James Allen, the principal investigator, received the Young Presidential Investigator's award, and Henry Kautz (Ph.D. in 1987) received the Computers and Thought award.

1.3.10 Inference Techniques for Knowledge Base Maintenance using Logic Programming Methodologies (SU): Vol. 11

This project was concerned with the development of logic programming-based machinery for the management of large complex knowledge bases of a highly dynamic character, together with the development of mathematical foundations for such systems. Knowledge base management includes the maintenance of ordinary integrity constraints as well as sophisticated reason maintenance systems. The work was carried out from the point of view of certain *meta-level* extensions of Prolog, generically baptized *metaProlog*. The primary tasks of the project included the following:

- Continued development of the metaProlog system. The principal goals here are the construction of an efficient metaProlog compiler, development of sophisticated memory-management methods, the development of suitable interfaces to non-metaProlog external databases, and the study of co-routining and concurrency.

- Development of knowledge representation formalisms in metaProlog, including analogs of frames, semantic nets, etc.

- Study of the expression of generic database management and knowledge base reason maintenance approaches in metaProlog, with special attention being devoted to maintenance of static and dynamic integrity constraints, reason maintenance, and daemons.

- Construction of one or more experimental demonstration systems using the machinery developed.

- Exploration of semantic foundations for both classical logic programming as well as non-standard approaches showing potential for dealing with the theoretical problems which arise in knowledge base maintenance.

We developed considerable knowledge of the structure and uses of the metaProlog system, ranging from its theoretical underpinnings to its use for implementing such programming constructs as frames, semantic nets, and message passing. We also developed considerable expertise and tools concerning the implementation of systems of the character of Prolog and metaProlog. We first applied this to the construction of a byte-code interpreter-based compiler for Edinburgh Prolog which achieved 10K LIPs running the standard benchmark on a VAX 780. At the time, this was the fastest implementation of Prolog on the VAX 780. This was used to compile our first substantial simulator for metaProlog (written in Prolog), producing a system which enabled us to begin serious metaProlog-based experiments. We then began extensions of the abstract machine underlying the Prolog byte-code interpreter aimed at producing an abstract machine suitable for the compilation of metaProlog. We explored a number of alternatives which presented themselves, eventually consolidating most of the valuable ideas into one system. Two alternative approaches to one aspect of the system led to the development of two alternative versions of the metaProlog compiler system. Both versions implement the core metaProlog features:

- Theories (logic databases) as first-class program objects which can be the values of variables and returned by procedures;

- Direct program access to the underlying proof predicate.

Both are incremental and interactive compilers which appear to be interpreters, but which generate (very quickly) byte-coded instructions for the underlying abstract metaProlog machine; these instructions are executed by an abstract machine interpreter coded in C. (Following the pattern for ordinary Prolog, extremely efficient native code compilers can be developed from this architecture). Both systems had approximately the same efficiency as our earlier byte-coded Prolog compiler: approximately 8-10,000 LIPS on the native reverse benchmark depending on cache interaction, on a VAX 780.

Both systems were later extended to incorporate the following:

- Complete garbage collection. This included collection of compiled program code which is stored on the system heap.

- Proofs of goals as first-class objects. Consequently, programs can reason about the proofs resulting from solutions of goals (e.g., for explanation generation or for sophisticated fault diagnosis).

Very early on, we completed the axiomatization of a medium-scale knowledge base problem in Edinburgh Prolog. We used this experience to guide some of the investigations into the design of the metaProlog engine. Later, we converted it to run in the metaProlog system.

We also conducted an extensive study of the truth and reason maintenance literature, eventually focusing primarily on deKleer's Assumption-Based Reason Maintenance. Implementation of ABRM can be carried out using metaProlog. However, because of the logical character of deKleer's work, we studied methods of abstracting its basic facilities and directly incorporating them in metaProlog as system facilities.

We also constructed an interface from our original Prolog compiler to the academic version of the INGRES DBMS, and experimented with it. Because of the monolithic character of INGRES, communication between the two systems was limited to string-based communication, and the results were initially somewhat disappointing. However, we later ported the interface to the commercial version of the INGRES DBMS, and achieved much better results. This reinforced our conviction that there must be as close as possible communication between the Prolog/metaProlog system and any external DBMS system with which it is linked.

We examined a number of semantic approaches to clarifying the foundations of metaProlog. The fundamental difficulties arise from the "amalgamated character" of the language, wherein the variables of the language must not only range over conceptually ordinary individuals, but also over the syntactic constructs of the language itself, noting that the language is untyped (like LISP and ordinary Prolog). Several directions explored included using "possible world" semantics and a semantics in which ordinary logical structures interpreting the language are extended to include abstract syntactic entities generated (rather like a word algebra) from the individuals of the interpretation.

The first approach, while intuitively appealing, does not seem to lead to useful tools. The second approach has promise, but does seem to entail considerable complexity. However, a third approach (which is definitely related to the second) suggested itself, and this seems to have even greater potential. In essence, this approach follows the so-called "substitutional interpretation" of logic, but instead of basing the work on the traditional two-valued truth values, utilizes collections of partial search spaces for proofs in the language as a set of truth values.

We devoted considerable effort to exploring theoretical approaches to default reasoning, inconsistency, stratified knowledge bases, non-standard logics, topological semantics, and multi-valued logic programming. These investigations were quite successful. The details are presented in the body of the report.

1.3.11 Computer Architecture for Very Large Knowledge Bases (SU): Vol. 12

The focus of our research was on the development of algorithmic, software and hardware solutions for the management of very large knowledge bases (VLKB) in a real time environment. We approached the problem from electronic and optical points of view. The electronic approach was based on more traditional digital computer technology, and we developed algorithmic and hardware solutions to the VLKB problem. We assumed a logic programming inferencing mechanism and a relational model for the management of the knowledge base. The interface between the inferencing mechanism and the extensional database becomes one of partial match retrieval. We bridged the gap between the two parts through the use of a surrogate file structure for the representation of both rules and facts.

In the optical approach we were concerned with the high speed and massive inherent parallelism of optics and how they might be used to advantage in storage, transport and processing of very large knowledge bases.

In the general case a logic programming front end engine requires equal access to all rules and facts. Because of this generality we took a surrogate file approach to the management of the VLKB. Surrogate files are transformations that yield improved performance because of smaller size, more rigid structure and the opportunity for parallel operations. In prior work we analyzed several possible surrogate file structures and selected concatenated code words (CCW) as the approach that offered the most generality and potential performance improvements. Basically, a CCW is a concatenation of transformed values and one can utilize the individual components of the CCW as well as the entire word.

We designed a parallel back end database machine. The basic idea of the machine is to reduce the amount of fact data transferred from the secondary storage system while satisfying the user query. In order to do this the CCW files are distributed over many disks which are under the control of many surrogate file processors. The CCW entries are used to greatly reduce the amount of data that needs to be searched in response to a query. Relational data base operations are performed on the surrogate files thus further reducing the amount of data needed to be retrieved. When all operations are complete the results are then sent to the logic programming engine for further use.

Another important advantage of the CCW surrogate file technique is that it can be used for the indexing of rules expressed as logic programming clauses, where the matching between constants, variables and structured terms is required to test for unifiability. The CCW is obtained from the arguments in the clause as well as the predicate name of the head of the clause. Each code word is divided into a tag field and a value field. The tag field can represent any argument type including lists, structured terms, variables and constants. The value field contains the transformed representation of the corresponding argument according to the content of the tag field. Thus, the CCW approach allows for the representation and processing of rules and facts in a unified manner.

We analyzed the CCW technique in a variety of ways, including simulation on the Connection Machine and the development of a demonstration system. The demonstration system consisted of Prolog, INGRES and specially developed modules. The system allowed for the generation and management of surrogate files of various types, the execution of Prolog programs and the management of rules and facts.

To handle very large dynamic databases we have developed the dynamic random sequential access method (DRSAM). It is based on an order preserving dynamic hashing method derived from linear hashing. The performance of DRSAM was evaluated and found to be efficient for range queries as well as random access. With order preserving hashing, the hashed key values are not generally uniformly distributed over the storage address space. To deal with the nonuniformity we extended DRSAM with additional control structures.

The use of optics in the management of VLDB's can be divided into three parts: storage, transport and processing. Storage involves the use of optical disks or holograms. It appears to be feasible to obtain at least two orders of magnitude increase in optical disk transfer rates through the use of multiple beam reads. These data could be input to optical fiber for transport to optical database processors. We developed an initial design of a system for the performance of various VLKB operations. It can perform selection, projection and equijoin as well as the filtering of ground clauses. The configuration includes two spatial light modulators and a large photodetector array for photon/electron conversion.

1.3.12 Parallel Vision Systems; Image Understanding and Intelligent Parallel Systems (UofR): Vol. 13

Rochester recognized early on the importance of parallel computation in all its technical manifestations to the problem of computer vision. Aided significantly by NAIC support, we have explored several different paradigms for parallel vision and have made significant contributions in the area. This funding took the form of two technical tasks: the first in Parallel Vision Systems, and the second in Image Understanding. Owing to the proximity of these research areas, we have included the results in a single volume.

1.3.12.a Parallel Vision Systems

The individuals funded in this work include Chris Brown, Paul Cooper, Michael McInerny, Paul Chou, and Sara Porat, although many more workers have been attracted into this exciting field, so that the NAIC support has been highly leveraged.

When funding commenced, Rochester was getting started in two main research directions which have since proved to be highly successful, and during the funding period several new initiatives have been undertaken.

The oldest central themes in our parallel vision work are those of massively parallel (connectionist, neural net) computation, and MIMD (multiple independent processors) computer hardware and the software to make it useful. Early work involved SIMD-like (data) parallelism on the BBN Butterfly parallel processor (Rochester had the largest Butterfly configuration outside Washington D.C at the time), investigation of multi-process scheduling for object tracking, the development of the Rochester Connectionist Simulator (since distributed to hundreds of sites worldwide), and theoretical studies on pipelined, MIMD, and massively parallel computation. The development of optimal low-level feature detectors using Bayesian probabilistic models was a high priority.

Following on the Bayesian feature detector work was work by Paul Chou, who used Markov random fields for sensor fusion and multi-modal segmentation (using intensity and sparse depth information). Following on from Chou's work was that of Cooper and Swain, who used Chou's formalism in developing algorithms for generic parameterized object detection using neural nets, and designed hardware to accomplish the work. The Markov random field work also led to work in parallel evidence combination.

At about this time our parallel computation facilities were enhanced by the MaxVideo pipelined image processor. This device made real time vision possible, and developing algorithms for it has been a continuing activity. Our complement of boards has grown from six to about twenty, in two card cages, and this form of parallelism has become an important part of our scientific culture. The MaxVideo system is a central component in the integrated robot system work (Image Understanding and Intelligent Parallel Systems) also supported by NAIC. Studies carried out on this architecture have been reported in many major refereed conferences, including International Joint Conference on AI 1989 ("best paper" award), Computer Vision and Pattern Recognition 1989 (two papers), International Conference on Computer

Vision 1988, NATO Workshop in Active vision and robotics 1989 (two papers), and several more.

Object recognition was approached not just for objects with much relational structure but for polyhedral objects seen from general viewpoints. Watts' algorithm for generating all distinct views of a class of nonconvex polyhedra is a significant contribution to the literature in this area.

Sara Porat formalized, proved, and published (in Biological Cybernetics) results in the convergence and power of certain classes of parallel neural net computations.

At about this time Randal Nelson was hired from U. Maryland. He is a student of Rochesterian J. Aloimonos. He completed work on a navigation system that used a distributed representation based on local clues rather than a global map to find its home base. This year has developed a real-time, MaxVideo based optic flow algorithm. Paul Chou received his Ph.D. and went to IBM T.J. Watson Research Labs, and Paul Cooper also graduated and is now at Roger Schank's AI Institute at Northwestern U. Brown, while on sabbatical, produced several papers, conference talks, seminar talks, and technical reports in three main areas: Kalman filtering applied to target tracking, algorithms for parallel cooperating controls for managing gaze in an advanced vision system, consideration algebraic and geometric projective invariants in vision and their practicality for overcoming viewpoint variation, and an in depth study of the computational properties of rotation representations.

Vision remains an intractable but central problem, and much remains to be done to apply current parallel technology effectively to vision. Current work involves using MaxVideo to compute real-time color histograms as input to a neural net that learns object and location recognition, the planning of foveation sequences (where to look next), testing various models of primate color vision and gaze-control functions, color segmentation and object recognition using color, and several neural net projects. The scientific coupling between parallel vision, parallel systems, and robotic effectors remains strong, as does work in basic parallel vision algorithms. The PLATINUM system automatically solves the problems isolated by Tom Olson in his earliest work on SIMD vision algorithms on the Butterfly, thus bringing that work full circle. The Psyche operating system for the butterfly will let us apply, under one operating system, whatever sort of parallel computation (MIMD, SIMD, shared memory, message passing) is most appropriate to the problem. NAIC's substantial contribution has been multiplied by other resources, and has been instrumental in supplying the impetus to propel Rochester into the front ranks in this research area and the continued support to maintain our viability during the sometimes confusing periods of growth.

1.3.12.b Image Understanding and Intelligent Parallel Systems

With the help of NAIC support, Rochester has been following a course research that builds on its long standing interest and achievement in parallel vision (also funded by NAIC). The individuals directly funded in this work include Chris Brown, Randal Nelson (a newly-acquired faculty member attracted by this area of research) Tim Becker, David Coombs, Ray Rimey, and Brian Marsh, although the high excitement level generated by the work has attracted many more workers into the area, so that the NAIC contribution has been highly leveraged.

Systems that behave in the world are becoming increasingly sophisticated, raising technical problems of sensing and control and opening new approaches that may make perception easier. One of the goals of artificial (animate vision), enunciated in work by Rochesterians Aloimonos, Bandyopadhyay, Brown, Nelson, and Ballard, among others, is to exploit the ability to maneuver in 3D to make some vision problems easier. Another goal is to design a systems architecture in which multiple objectives (such as moving and observing) can proceed in parallel. One common premise is that cognitive processes at high abstraction levels rely on a hierarchy of lower-level "reflexes" or "behaviors", and active vision capabilities that autonomously keep the agent out of trouble and perform generally useful vision computations. Another premise is that active control over and perception of an agent's own state (proprioception) makes many problems in perception, planning, and acting easier.

Implementing a system to demonstrate these principles has involved us in the purchase of a powerful robot arm, in the design and fabrication of a three degree of freedom binocular robot head, and in the attendant issues of systems building and applications programming. Studies carried out on this architecture have been reported in many major refereed conferences, including International Joint Conference on AI 1989 ("best paper" award), Computer Vision and Pattern Recognition 1989 (two papers), International Conference on Computer Vision 1988, NATO Workshop in Active vision and robotics 1989 (two papers), and several more.

This work is supported by the Parallel Vision research effort supported by NAIC, which develops algorithms for parallel computer architectures, be they simulated parallel systems (neural nets, Markov random fields) or actual MIMD parallel computers (the BBN Butterfly) or pipeline parallel computers (the Datacube MaxVideo system).

The IU and PS effort is especially lucky to have captured the imagination of the strong Computer Systems Research Group at Rochester, and significant cooperation is taking place. The CSRG, which consists of three professors and some eight students, is developing Psyche, an operating system with a real-time kernel to run the 24 processor Butterfly-II computer on active vision and robotics tasks, which ultimately will take over the host function for the arm and head. In other work they are developing languages, debugging and performance monitoring tools to make programming MIMD computers possible. The first application robotics and vision task is to keep a balloon in the air, which combines aspects of real-time image analysis and control.

This work is evolving in four separate directions, which NAIC has been crucial in starting. Increasing functionality for world interaction: in particular, using an advanced MIT-Utah hand for skilled eye-hand tasks. Integration with high-level symbolic planners: in particular, visual inspection of a model train layout integrated with symbolic planning to achieve goal configurations. Integrating low-level functionalities: in particular, creating a gaze control system to manage the competing and cooperating demands on the vision system. Integrating the vision and robotics system with modern parallel hardware. The work will continue to expand and flourish, and we expect to keep the lead in this work that NAIC has helped us establish.

1.3.13 Knowledge Base Retrieval using Plausible Inference (UMass): Vol. 14

One of the primary functions in many knowledge-based applications is the retrieval of objects that satisfy criteria specified in a user's query. Examples of objects that may be retrieved in this way include natural language documents or parts of documents, and fragments of encyclopedic knowledge bases. In these cases, the process of determining if a query criterion is satisfied will involve inference. One approach to this problem would be to represent objects and knowledge about objects in a deductive database system. In such a system, a query can be expressed in the form $q = \{X \mid W(X)\}$ where X is a vector of domain variables and $W(X)$ is a formula in which X are the only free variables. Retrieval involves finding all instances of X for which $W(X)$ can be proved. In other words, for a query q, retrieve X where $KB \models W(X)$. The knowledge base, KB, includes descriptions of objects (extensional data), rules (intensional data), and basic axioms. The main issue in implementing deductive database systems is designing efficient inference methods.

The critical issue we have studied is the *effectiveness* of retrieval. By this, we mean how well the system does at locating objects that are judged relevant by the user. Designing effective retrieval strategies is difficult because in real environments the query specification, the object descriptions, and the rules in the knowledge base are incomplete and uncertain. In these situations involving uncertain information, *deductive inference does not provide effective retrieval.* Instead, retrieval must be implemented as a process of plausible inference or evidential reasoning.

The aim of our research is to demonstrate that plausible inference is an effective computational framework for retrieval of complex objects. To do this, we must address a few key issues. In particular, we must describe an appropriate formal model of plausible inference, show how this model is implemented in a real system, and evaluate its performance. In the relatively short duration of this project (14 months), we have begun to make significant progress in all these areas.

In particular, we have pursued both numerical and symbolic approaches to representing and reasoning with uncertainty. Recently, we have begun investigating a Bayesian network approach due to the fact that simpler probabilistic models have produced good results in previous research.

We have also been constructing an experimental setting for our research with test collections that consist of a large number of text objects, a linguistic knowledge base, and a domain knowledge base that describes objects and relationships in the domain of texts. These test collections also must have associated queries and relevance judgements. Some retrieval experiments using heuristic versions of the plausible inference model have been carried out with encouraging results.

We have also concentrated on developing techniques for automatically producing complex representations of the meaning of text objects. These natural language processing techniques provide rich sources of evidence for our plausible inference models, as well as providing the basis for important applications of text-based intelligent systems.

Finally, we have conducted a number of experiments on the derivation of plausible inference rules from knowledge bases.

1.3.14 Strategies for Coupling Symbolic and Numerical Computation in Knowledge Base Systems (SU): Vol. 15

It is widely recognized that coupling symbolic and numerical methods is an effective means of solving many problems in science, engineering, and business. There are two major reasons for the recent interest in coupling numerical and symbolic processes. The first reason is the need to help a problem solving process that needs to use complicated numerical algorithms and programs. In order to solver various problems in science, engineering, and business, both insight and precision are very frequently needed. Coupled systems guarantee the integration of the precision of traditional numerical computing with the problem solving and result explaining capabilities of symbolic processing. The second major reason for coupling is the need for more powerful and useful tools capable of overcoming the limitations of traditional tools, which are not coupled systems. Integrating formal numerical methods with methods based on symbolic knowledge is believed to be the key to the development of computing tools that can solve some of the problems currently known to be difficult to solve.

Coupled systems link symbolic and numerical computing in a manner not found in conventional expert or knowledge-based systems. For coupled systems, effective problem solving must have some knowledge of the numerical processes embedded within them and must be able to reason about the application or results of those numerical processes. In general, the symbolic process of a coupled system is the top-level process controlling the numerical process. However, the possibility of a numerical process controlling symbolic processes cannot be neglected, although numerical algorithms are only able to procedurally invoke symbolic processes. It is possible for a numerical process to possess symbolic capabilities.

In order to study coupled systems in the context of computer vision, the problem of 3-D object recognition from range data in a multiple-object scene with partial occlusion was considered. This problem is of considerable theoretical and practical interest. This problem is encountered in several scenarios such as robot bin-picking, automated industrial inspection, autonomous navigation, etc. The two main problems encountered when dealing with a multiple-object scene are (i) combinatorial explosion of the search space of scene interpretations, and (ii) generation of spurious scene interpretations. Thus, the issues of *representation* and *constraint propogation / satisfaction* were dealt with primarily with the following objectives in mind: (i) reducing the combinatorial complexity of the search space of possible scene interpretations and (ii) ensuring robustness against occlusion.

Our work thus far has brought out the advantages of using qualitative features in achieving these objectives. With Hough clustering as the chosen constraint propogation / satisfaction technique, three problem scenarios of increasing complexity were used to demonstrate the effectiveness of using qualitative features for recognition and localization: (i) recognition of polyhedral objects, (ii) recognition of curved surfaces in particular conical, cylindrical, spherical, and planar surfaces, and (iii) complex objects made up of piecewise combinations of conical, cylindrical, spherical, and planar surfaces. The results are analyzed both in terms of accuracy and robustness against occlusion. The choice of Hough clustering as the constraint propogation / satisfaction technique was governed by the fact that it is easily amenable to parallelism.

This research clearly shows the role of qualitative features in recognition and localization. Qualitative features provide:

- An effective means of reducing the combinatorial complexity of the search space of possible scene interpretations. In the context of Hough clustering this translates to being able to suppress spurious peaks in the Hough space.

- An effective criteria for choosing the appropriate representation for the image data (scene features), object models (model features), and constraints resulting from matching scene features to model features. In the context of Hough clustering this translates to choosing the appropriate parameter space in which to compute the object pose.

The advantage of using qualitative features for recognition and localization was brought out through our experiments in all the three forementioned problem scenarios. The use of qualitative features was shown to greatly enhance the performance of conventional Hough clustering in terms of both (i) robustness of the recognition process and (ii) accuracy of the localization process.

The use of qualitative features can be seen as a criteria for selection of the appropriate granularity of representation and constraint propogation. Thus, qualitative features would serve as an indexing criteria in a coupled systems approach to problem-solving in computer vision.

The work done so far has brought out the advantages of using qualitative features as a means for indexing into appropriate representations and method selection for three-dimensional object recognition. The future directions for research can be briefly outlined as follows:

- Implementation of an object-oriented framework for 3-D object recognition in C++ or CLOS;
- Tackling other vision problems in the framework of coupled systems, namely motion, texture, and stereo;
- Parallel implementation of constraint propogation / satisfaction algorithms for vision problems.

1.3.15 Intelligent Signal Processing for Surveillance Systems (RIT, RPI, SUNYAB, UofR): Vol. 16

Current surveillance systems must perform a variety of tasks within a complex real-time environment. Artificial intelligence (AI) techniques which have been developed for modern signal processing applications such as vision, image understanding and speech understanding combined with other AI techniques such as expert systems, knowledge representation, plan recognition, search and control offer ways to improve the performance of the next generation of surveillance systems.

It is likely that the next generation of surveillance systems will combine the information from sensors with knowledge from other sources in synergistic ways to provide for better performance in detection, tracking and threat recognition. The information from distributed sensors and processors will be combined in processors which can utilize high-level symbolic algorithms.The approach offers the opportunity for modularity in system development, deployment and maintenance.

If effective custom systems can be constructed from generic modules then there will be a very substantial savings over the cost of individual custom systems for a variety of applications.Intelligent systems offer systematic processes to address problems such as sensor fusion, sensor coordination, threat assessment, decision analysis and resource allocation. Such tasks require that information be handled at a high level so that symbolic reasoning can be supported.

The behavior of distributed systems with significant numbers of interacting components is difficult to analyze and predict. Even if the individual components are well-understood, a system composed of many of them may exhibit new and unexpected behavior. The inclusion of nonlinear processes, as decision processes must be, makes the theoretical analysis of such systems essentially impractical. Therefore the behavior and performance of distributed systems can be best evaluated by the use of prototypes and simulation.

The purpose of this project is to develop an analytical framework to represent the modern multi-target, multi-sensor surveillance environment and to investigate the adaptation of intelligent signal processing algorithms to that application.The first year of this study developed a road-map for research and development in this arena. The current effort has been devoted to the development of a simulation test-bed and the development of basic simulation representations and algorithms.

SIMULATION SYSTEM

Under the current phase of the study a simulation system has been implemented on a TI Explorer II in TI Common Lisp with Flavors. The simulation features an object-oriented approach in which such components as transmitters, receivers, signal processors, targets and the signals themselves may be represented.Class inheritance provides a powerful mechanism for defining new classes whose behavior is similar to existing classes.

The advantages of the object-oriented approach are:

1. Easier and more rapid prototyping - essential to implement and maintain large systems.

2. Data encapsulation - which encourages and supports good programming practices.

3. Networking and concurrency issues are more easily handled.

4. Code reusability is supported.

The simulator provides a life-cycle description of each object in a method. This provides an intuitive and flexible way to define and observe objects in a simulation. Objects interact by passing messages and taking actions on the basis of the messages that they receive.

Simulations are designed and executed through a graphical user interface. Icons representing instances of objects are placed on the screen through a mouse and menu system. Paths are drawn to indicate logical connections among objects. Messages and object movements are illustrated by animation during execution. Each object may have several windows in which it can display its activities. Objects can be nested hierarchically either to group a collection of objects or to define a functional operation. Simulation may be interrupted, modified and resumed.

STATUS

The software shell for the simulator has been essentially completed. It supports the functionality described above. However, the simulator has not been populated with typical objects and an advanced dataflow structure for the representation has not yet been implemented. The simulator shell is capable of representing complex environments, but does not yet contain algorithms for performing information extraction and symbolic reasoning. These additions will be constructed on the basis of research to be conducted in future phases of the effort.

1.3.16 A Knowledge Acquisition, Systems, and Explanation System: High-Level Adaptive Signal Processing (UMass, Boston Univ.): Vol. 17

In this report, we describe the accomplishments of the high level adaptive signal processing (H-LASP) project, carried out by a team of researchers from University of Massachusetts at Amherst and Boston University during the period from February 1989 to September 1989. High-level adaptive signal processing (H-LASP) involves the integration of artificial intelligence and signal processing in an interpretation system and makes use of a paradigm that allocates processing resources and adjusts parameters of the low-level processing in accordance with the evolving high-level interpretations of the signal-generating environment. The goal of the project reported here was to evaluate how the H-LASP paradigm applies to a realistic task: real-time sound classification. We have built a testbed for this application and found that with some modifications and a number of refinements, the H-LASP paradigm can be successfully used for the development of signal interpretation systems.

In high-level adaptive signal processing, the integration of high and low level processing is achieved through a problem-solving paradigm that involves three phases: discrepancy detection, diagnosis, and parameter adjustment. Discrepancy detection is carried out by comparing the features of the signal processing outputs with features expected on the basis of the evolving scenario interpretation and apriori knowledge about the signal-generating environment. This is followed by a diagnostic reasoning process that makes significant use of the underlying Fourier theory of the signal processing system to isolate a subset of system parameters whose settings were likely to have caused the observed discrepancies. Finally, the signal processing resources are reallocated by appropriately adjusting system parameters with the aim of removing the observed discrepancies. This paradigm was established in our previous research on an acoustic localization problem where we had found that expert human signal processors use this type of reasoning in manually reallocating the signal processing resources through parameter adjustment. The need for resource allocation for the low-level processing components arises because of two factors. The *signal variety* factor is that the signal processing resources (which are always finite) have to deal with an infinite variety of signal classes whose signal processing requirements are often in conflict with each other. By adjusting the parameters of an algorithm it can be made to deal with different classes of signals. The second factor that leads to the need for resource allocation is the *real-time performance* factor. In a real-time situation, there is not always enough time to do all the signal processing the system would ideally carry out. In such cases, focus-of-attention decisions have to be made about the use of the signal processing resources within the available time frame.

For the project described in this report, the goal was to evaluate and improve the H-LASP paradigm for a practical sound classification application. We selected the real-time sound classification problem for this purpose because it offers two major advantages: (1) it shares many low-level and high-level processing requirements with other signal interpretation problems such as radar signal interpretation and (2) the acoustic signal database is readily available in our university laboratories for testbed experiments. The specific sound classification problem arises in the context of real-time interpretation of acoustic signals received by a system (robot, if you will) stationed in a household environment. This means that the various sounds being received by the system have to be classified in terms of the sources from which these sounds originate. In the household environment, we are interested in sources such as telephones, vacuum cleaners, babies, speech, footsteps, doorbells etc. The problem is made particularly complicated (thereby requiring Artificial Intelligence techniques at the higher levels) because several sources may occur simultaneously and they may have overlapping frequency spectra.

The achievements of our project may be divided into five major categories:

- Incorporation of the diagnostic reasoning process into the sound classification testbed along with refinements in that process to deal with the more sophisticated theory underlying the new application.

- Formulation and implementation of a practical approach to discrepancy detection for the sound classification task.

- Implementation in the testbed of a sophisticated database using the Generic Blackboard (GBB) system. The design of the database within a blackboard framework was found to ease the development of the processing components of the H-LASP paradigm in the form of independent knowledge sources.

- Design of the control component of the testbed through adaptation of a framework developed at the University of Massachusetts for the control of interpretation through analysis of the sources of uncertainty associated with the various evidence gathering mechanisms.

- Design of the control component of the testbed to ensure real-time invocation of the high and low-level knowledge sources while maintaining the integrity of the high level interpretations to within the goals of the system.

Within its limited eight-month duration, the project was successful in developing a testbed that includes a blackboard database with knowledge sources for signal processing, discrepancy detection, and diagnosis. The parameter adjustment and system control components were fully designed, however, further work is needed to complete the implementation of the parameter adjustment knowledge sources and the control component of the system. Completion of these components will permit us to thoroughly evaluate the performance of a fully integrated H-LASP system for a practical real-time signal interpretation application.

1.3.17 A Distributed Artificial Intelligence Approach to Information Fusion and Object Classification (SU): Vol. 18

This project presents an application of Distributed Artificial Intelligence (DAI) tools to the data fusion and classification problem. The domain of application is modern C^3I systems where accurate and timely perceptions of friendly and hostile forces are required for effective decision making and battle management. DAI provides a means for interconnecting multiple expert systems that have different, but possibly overlapping expertise, thereby enabling the solution of problems whose domains are outside that of any one expert system or knowledge source.

The approach is to use a blackboard for information management and hypotheses combination and formulation. The blackboard is used by the knowledge sources (KSs) for sharing information and posting their hypotheses, just as experts sitting around a round table would do. The present simulation performs classification of an aircraft (AC), after identifying it by its features, into disjoint sets (object classes) comprised of the five commercial ACs: Boeing 747, Boeing 707, DC10, Concord, and Boeing 727. A situation data base is characterized by experimental data available from the three levels of expert reasoning. Ohio State University ElectroScience Laboratory provided this experimental data. To validate the architecture presented, we employ to KSs for modeling the sensors, aspect angle polarization feature, and the ellipticity data. The system has been implemented on a Symbolics 3645, under Genera 7.1, in Common Lisp.

1.4 ANCILLARY GOALS OF THE CONSORTIUM

The ancillary goals of the Consortium can be described as increasing the level of expertise in artificial intelligence at three levels: internal to the Consortium; between Consortium members and other entities (academic and commercial); and in the community at large.

1.4.1 Development of Internal AI Expertise

INTERACTION BETWEEN CONSORTIUM MEMBERS

The principal method of developing internal AI expertise is by fostering interaction between members of the Consortium. The NAIC fostered a tremendous amount of interaction, which includes collaborative research, regular meetings of the Executive Committee (every six weeks), the Annual Meeting, "focus" meetings, workshop series, inter-school visits, etc. Where appropriate, proceedings, notes, and attendee lists have been published as technical reports by the Consortium, enhancing further the opportunities for intellectual exchange.

As can be seen from the list of research projects, research collaborations sprang up between researchers at most of the institutions. Other indications of the quantity and quality of interactions can be seen through the number and variety of formal gatherings of the Consortium. The following formal meetings were held over the five years:

Spring Meeting: "Plan Recognition"
March 12 - 13, 1986
University of Massachusetts
Amherst, MA

Annual Meeting
July 9 - 11, 1986
University of Rochester
Rochester, NY

Fall Meeting: "Spatial Knowledge Representation and Reasoning"
September 25-26, 1986
State University of New York at Buffalo
Amherst, NY

RADC/NAIC Technical Fair
April 9 - 10, 1987
RADC
Griffiss AFB, NY

Annual Meeting
June 29 - July 1, 1987
Minnowbrook Conference Center
Blue Mountain Lake, NY

"Natural Language Planning"
September 20 - 23, 1987
Minnowbrook Conference Center
Blue Mountain Lake, NY

Fall Meeting: "Planning"
August 2, 1987
Clarkson University
Potsdam, NY

Spring Meeting: "Vision and Intelligent Signal Processing"
March 29-30, 1988
Hyatt Regency, Crystal City
Washington, DC

RADC/NAIC Day
May 11, 1988
RADC
Griffiss AFB, NY

Rockwell/NAIC Artificial Intelligence Symposium
June 6-7, 1988
Sheraton Conference Center
Syracuse, NY

Annual Meeting
August 8-11, 1988
Minnowbrook Conference Center
Blue Mountain Lake, NY

Planning Workshop: "From Formal Systems to Practical Systems"
October 27-29, 1988
University of Rochester
Rochester, NY

Spring Meeting: "Neural Networks & Complex Distributed Systems"
April 13 - 14, 1989
Rochester Institute of Technology
Rochester, NY

Annual Meeting
August 14 - 17, 1989
Minnowbrook Conference Center
Blue Mountain Lake, NY

FACULTY AND GRADUATE STUDENT PARTICIPATION

The level of participation by faculty and graduate students at each school can be viewed in two ways: the sheer numbers of participants, and the growth that took place in each institution as a result of its participation in the Consortium. It is this latter aspect that best captures the original intent of the NAIC; however, each view offers substantial evidence of success.

SUNY at Buffalo saw seventeen graduate students supported by the NAIC. Ph.D.'s awarded per year in AI grew from none in 1984 to five in 1988, for a total of 11. Of nine AI faculty, five were added during the period of the NAIC. Currently, ten Ph.D.'s are in progress.

Clarkson University saw three Master of Science degrees completed (with another three underway) and two Ph.D.'s (with one more underway) as a result of NAIC involvement. Furthermore, three new AI faculty will have joined the department by the end of this year.

At the University of Massachusetts, 34 students completed their Master of Science degrees, 22 of whom went on to pursue their Ph.D.'s. Of these 22, three finished their doctorates, and seven more are expected to complete their degree by the end of this academic year. Research faculty grew by a factor of 141% over this same period.

At the University of Rochester, five students completed their Master of Science degrees, another four completed their Ph.D.'s, and eight more Ph.D. candidates are underway.

At the Rochester Institute of Technology, 13 Master of Science degrees were awarded (RIT has no Ph.D. program); and four faculty and two full time research staff were added.

Similar growth was enjoyed at Rensselaer Polytechnic Institute and Syracuse University, although detailed statistics are lacking. No such growth could be identified at either Colgate or Boston University, owing to their peripheral involvement.

NEW COURSES AND INSTITUTIONAL RESOURCES

Every institution saw a direct enhancement of its institutional AI resources as a result of its involvement in the NAIC. Owing principally to the efforts of Robert Meyer and Susan Conry at Clarkson University, a URIP equipment grant of $250,000 was obtained, which was used to purchase sixteen Texas Instruments Explorers. This equipment, which was distributed throughout the NAIC, served as a foundation for research software development, and allowed true collaboration to proceed. Eventually, the efforts of the Managing Director led to the donation of another twelve Explorers to Consortium members.

Each institution was able to increase its resources beyond these specific acquisitions. At the University of Massachusetts, for example, their AI computing resources were increased to 72 Lisp machines and a Sequent Multiprocessor; RIT acquired additional workstations (including Suns, Macintosh's, and Micro-Explorers); SU acquired a MicroExplorer, and (in a related development) acquired a great deal of parallel processing capability (two Connection Machines, two Encore Multimaxes, and one Alliant), with one of the principal research foci being artificial intelligence. Again, similar growth is reported by other member institutions.

In terms of course development, again, the number of courses and research areas in artificial intelligence was significantly expanded. Clarkson added six courses, UMass added 19, RIT added four (plus eight "special topic" courses), Syracuse and SUNY Buffalo added three.

1.4.2 Development of AI Expertise between Consortium Members and Other Entities

1.4.2.1 COMMERCIAL PARTICIPATION

During the course of the Consortium, relationships with industry were solidified, and productive arrangements were undertaken. These arrangements took the form of joint conferences, collaborative research, consulting arrangements, and, in the case of Texas Instruments, a donation of twelve Explorer Lisp Machines in support of NAIC research and education. The list of industries with whom new relationships were forged by Consortium members during this period is impressive. A partial list follows:

BBN
Bell Communications Research
Digital Equipment Corporation
Eastman Kodak Corporation
General Electric
IBM
IIT Research Institute
ITT Defense Communications
Kaman Sciences Corporation
Knowledge Systems Concepts
Lincoln Labs
Lockheed
Marine Midland Bank
Niagara Mohawk Power Corporation
OCLC
Olivetti
PAR Technologies
Philips Labs
Ricoh
Rockwell International
Singer Aerospace and Marine Systems
Symbolics
Texas Instruments
Thinking Machines Corporation
UNISYS
United Technologies Corporation
Xerox

1.4.2.2 GROWTH IN VISIBILITY

Over the years of the Consortium's life, steady progress was been made towards achieving a greater visibility in professional circles. For example, in 1988, the NAIC achieved a 175% growth in meeting attendance over the previous year, a 36% increase over meetings held, and a 32% increase in articles published. The number of articles published and conferences attended is truly staggering; overall, well over two hundred national and international professional meetings were attended, and close to five hundred papers were published. Book chapters and books were produced at every institution as well. A partial list of this latter category are:

Allen, J., *Natural Language Understanding*, Benjamin Cummings, 1987.

Allen, J.F. and Kautz, H., "A Model of Naive Temporal Reasoning," in *Formal Theories of the Commensense World*, J. Hobbs and R. Moore (eds.), Ablex Publishing Co., 1985.

Allen, J. Chapters in: *Computational Models of Discourse*, (1984), *Contributions in AI*, (1985), *Statistical Image Processing and Graphics*, (1986), *Readings in Knowledge Representation*, (1985), *The Role of Intentions and Plans in Communications and Discourse*, (1988).

Brown, C. (ed.), *Advances in Computer Vision*, 1986.

Brown, C. Chapters in: *3-D Representations and Descriptions* (1984), *Statistical Image Processing*, (1986), *Range Data*, (1988).

Brulé, J.F. and Blount, A., *Knowledge Acquisition*, McGraw-Hill, 1989.

Conry, S.E., Meyer, R.A., and Lesser, V.R., "Multistage Negotiation in Distributed Planning," in *Readings on Distributed Artificial Intelligence*, A. Bond and L. Gasser, (eds.), Morgan Kaufmann Publishers, 1988.

Croft, W.B. and Lefkowitz, L.S., "A Goal-based Representation of Office Work," in *Office Knowledge: Representation, Management, and Utilization*, Lamersdorf (ed.), Elsevier Science Publishers, 1988.

Croft, W.B. and Lefkowitz, L.S., "Knowledge-Based Support for Cooperative Activities," in *Readings on Distributed Artificial Intelligence*, A. Bond and L. Gasser, (eds.), Morgan Kaufmann Publishers, 1988.

Lesser, V.R., Pavlin, J., and Durfee, E.H., "Approximate Reasoning in Real-time Problem Solving," to appear in *Blackboard Architectures and Applications*, L. Baum, V. Jagannathan, and R. Dodhiwala (eds.), Academic Press, 1989.

Lesser, V.R., Whitehair, R.C., Corkill, D.D. and Hernandez, J.A., "Goal Relationships and Their Use in a Blackboard Architecture," to appear in *Blackboard Architectures* *and Applications*, L. Baum, V. Jagannathan, and R. Dodhiwala (eds.), Academic Press, 1989.

Meyer, R.A. and Conry, S.E., "Communications," in *Expert Systems and Artificial Intelligence: Applications and Management*, T.C. Bartee (ed.), Howard Sams & Co., 1988.

Nirenburg, S. and Lesser, V.R., "Providing Intelligent Assistance in Distributed Office Environments," in *Readings on Distributed Artificial Intelligence*, A. Bond and L. Gasser, (eds.), Morgan Kaufmann Publishers, 1988.

Shapiro, S. (ed.), *Encyclopedia of Artificial Intelligence*, John Wiley & Sons, 1986.

Woolf, B., "20 Years in the Trenches: What Have We Learned?" in *Intelligent Tutoring Systems: at the Crossroads of Artificial Intelligence and Education*, C. Frasson and G. Gasthier (eds.), Ablex Publishing Co., in press.

Woolf, B., "Hypermedia in Education and Training," in *Main Spring Symposium on Artificial Intelligence and Intelligent Tutoring Systems*, D. Kopec and B. Thompson (eds.), Ellis Norwood ltd., in press.

Woolf, B., "Intelligent Tutoring Systems: A Survey," in *Exploring Artificial Intelligence*, H. Shrobe and the American Association for Artificial Intelligence (eds.), Morgan Kaufmann Publishers, 1988.

1.4.2.3 AWARDS

Dr. Henry Kautz, an NAIC graduate from the University of Rochester, was awarded the 1989 "Computers and Thought" award, one of the most prestigious distinctions in the field. This award is given every two years by an international committee of scholars, who choose a researcher who has made significant contributions to artificial intelligence. His dissertation, completed in 1987 under Dr. James Allen, provided one of the first formal accounts of plan recognition.

Dr. James Allen (of the University of Rochester) was the recipient of a National Science Foundation Presidential Young Investor Award. Only a small number of such awards are presented each year, and competition for these awards is not just among AI researchers, but among researchers in each scientific discipiine.

The Encyclopedia of Artificial Intelligence, edited by Dr. Stuart C. Shapiro, professor and chairman of the Department of Computer Science at SUNY Buffalo, and published by John Wiley & Sons, won a first prize award as the Best New Book in Technology and Engineering for 1987 from the Association of American Publishers Professional and Scholarly Publishing Division.

Dr. S.N. Srihari with Dr. Jonathan Hull, research assistant professor of the Department of Computer Science at SUNY Buffalo, received third place award at the Niagara Frontier Inventor of the Year Award and Dinner on January 29, 1988. The award was for US patents issued to area inventors during 1987. Their patent is titled, "Systems to Recognize Bilinguistic Strings."

1.4.3 Development of AI Expertise in the Community

During the life of the Consortium, courses were offered for the professional community. Talks were also given to local groups of computer and managerial professionals interested in AI, such as to the Association for Intelligent Systems Technology.

MISSION

of

Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. The areas of technical competence include communications, command and control, battle management information processing, surveillance sensors, intelligence data collection and handling, solid state sciences, electromagnetics, and propagation, and electronic reliability/maintainability and compatibility.