ornl

ORNL/TM-9923
(CESAR-86/01)

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

Proceedings of a Workshop:

Planning and Sensing For Autonomous Navigation

E. S. Howe
C. R. Weisbin

CENTER FOR
ENGINEERING SYSTEMS
ADVANCED RESEARCH

CESAR

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A10 Microfiche A01

ORNL/TM-9923

CESAR-86/01

ENGINEERING PHYSICS AND MATHEMATICS DIVISION

CENTER FOR ENGINEERING SYSTEMS ADVANCED RESEARCH

PROCEEDINGS OF A WORKSHOP: PLANNING AND SENSING FOR AUTONOMOUS NAVIGATION

Edited by:

E. S. Howe and C. R. Weisbin

Date Published - January 1986

Research sponsored by
U.S. DOE Office of
Basic Energy Sciences

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
operated by
Martin Marietta Energy Systems, Inc.
for the
U.S. Department of Energy
under contract No. DE-AC05-840R21400

TABLE OF CONTENTS

ABSTRACT

This report represents the proceedings of the second DOE/CESAR Workshop entitled "Planning and Sensing for Autonomous Navigation." The meeting was held August 18-19, 1985 in conjunction with, and just prior to the International Joint Conference on Artificial Intelligence at the University of California, Los Angeles.

The workshop was organized around several issues developed to focus attention and clarify workshop priorities. The issues dealt with methods for "world mapping" and "discovery" in unstructured environments, approaches to real-time planning with sensor feedback, computer architectures and concurrent algorithms, sensor integration, and uncertainty representation and propagation. A series of overview papers contained herein served as background for discussion. Written summaries of group discussions were prepared during the meeting and are included in these proceedings.

INTRODUCTION

INTRODUCTION

The Center for Engineering Systems Advanced Research (CESAR) was established in 1983 at the Oak Ridge National Laboratory (ORNL) to address long-range, energy-related research in intelligent control systems. These systems are intended to plan and perform a variety of tasks in unstructured environments, given only qualitatively specified goals. The Center provides a framework for merging concepts from the fields of artificial and machine intelligence with advanced control theory.

In order to enhance cooperation with universities, laboratories, and industry, CESAR periodically organizes and conducts specialists' workshops. The first of these, held in Leesburg, Virginia on November 2-4, 1983, was the DOE/CESAR Workshop on Research Goals and Priorities in Intelligent Machines. A major accomplishment of this study was to identify those fundamental research areas that are not being addressed sufficiently by other organizations, and have a relatively high potential for medium- and long-range impact on the design of intelligent machines for energy-related environments. The proceedings of this workshop are available from CESAR upon request.

This report represents the proceedings of the second DOE/CESAR workshop entitled "Planning and Sensing for Autonomous Navigation." The meeting was held August 18-19, 1985 in conjunction with, and just prior to the International Joint Conference on Artificial Intelligence at the University of California, Los Angeles. Initial solicitations of interest were distributed in the January-March time frame; these were followed with official invitations for participation. The meeting attendance was kept relatively small (~30) to foster informality and free exchanges; however, the demand was such that a waiting list developed and not all interested parties could be effectively accommodated. The list of attendees, along with associated biosketches, is provided in Section V. They represent a broad cross section of university, laboratory, and industry.

The workshop was organized around several issues which were posed to the participants in advance of the meeting. These questions served to focus attention and clarify priorities. The issues were:

1. How can "world mapping" and "discovery" best be accomplished in unstructured environments?

2. What are the most promising approaches for real-time planning with sensor feedback from execution?

3. What are the most suitable computer architectures for the "brain" of an intelligent machine?

4. Can vision-dependent navigation in unstructured environments work effectively in real time; how is the information obtained best integrated with data from other sensors?

5. What are the most promising approaches toward uncertainty representation and propagation?

It was recognized from the outset that these questions are highly correlated; during the meeting it was decided to merge questions 2 and 4.

In order to effectively discuss these highly complex issues, a series of overview papers were presented. These papers, presented in full in Section II, are listed below:

1. Alberto Elfes: "Multiple Levels of Representation and Problem Solving Using Maps from Sonar Data"

2. Stan Rosenschein and Leslie Kaelbling: "The Synthesis of Digital Machines with Provable Epistemic Properties"

3. Jacob Barhen: "An Intelligent Machine Operating System for Hypercube Ensemble Architecture"

4. Scott Harmon: "Planning for Transit in Unknown Natural Terrain"

5. Ed Oblow: "O-Theory: A Hybrid Uncertainty Theory"

Elfes described his Dolphin system, a probability-based sonar map representation in multiple levels of resolution, used for successful indoor and outdoor navigation. Rosenschein and Kaelbling have adopted a design approach using epistemic logic in the formal analysis of the robot's information status, and using metaprograms that automatically construct real-time control programs amenable to this type of formal analysis. Barhen discussed an intelligent machine operating system based upon a virtual time paradigm including scheduling and load balancing the activities of multiple parallel processors. Harmon presented an approach for navigation in unknown natural terrain in which the route planning problem is subdivided into "orienteering" using domain specific knowledge and "global temporal planning" which is more problem independent. Finally, Oblow offered an uncertainty theory formulation based on the Dempster-Shafer approach and intended to bridge the gap between fuzzy set theory and Bayesian inference theory. Presentation of these five overview papers constituted the more formal part of the workshop.

Following the overview papers, each of our discussion leaders led the group in addressing the workshop themes listed above (recall questions 2 and 4 were merged). Section III contains the informal discussion summaries prepared by Jim Crowley, Stan Rosenschein, Jacob Barhen, and Peter Cheeseman. The discussion on each of these subjects was lively and interesting. VLSI and concurrent computation are needed for real-time robotic systems; more hands-on experience is needed for a specific architecture to be recommended. World modeling was seen as a problem dependent hierarchy ranging from wall (or road) following, to geometric modeling, object discoverers, and strategy learners in various degrees of sophistication. High-level and low-level planning processes may proceed at different time scales, but must be coordinated if timely response to environmental events is to guaranteed. This section also presents what is now fondly known as the "Cheeseman Challenge" and some of the correspondence related to consistent and comprehensive uncertainty analysis.

The group discussions were not intended to provide the final word on these complex issues. Rather the CESAR Workshop was intended to provide a useful forum for technical interchange on such important issues. Other written feedback from participants is also included in this section.

The positive feedback received from this meeting encourages us to begin planning our next meeting. We welcome your suggestions for recommended subject areas.

INVITATION

INVITATION

OAK RIDGE NATIONAL LABORATORY
OPERATED BY MARTIN MARIETTA ENERGY SYSTEMS, INC.

POST OFFICE BOX X
OAK RIDGE, TENNESSEE 37831

March 22, 1985

To: Workshop Participants: Artificial Intelligence and Mobile Robots

From: C. R. Weisbin, Director, Center for Engineering Systems Advanced Research (CESAR)

Subject: DOE/CESAR Workshop "Planning and Sensing for Autonomous Navigation" August 18-19, 1985

I. INTRODUCTION:

In behalf of the Department of Energy's Office of Basic Energy Sciences, the Oak Ridge National Laboratory Program in Intelligent Machines (CESAR) plans to held its 1985 technical workshop "Planning and Sensing for Autonomous Navigation," on August 18-19, 1985. This meeting should be of significant interest to attendees of the St. Louis Workshop on "Artificial Intelligence and Mobile Robots," since both meetings concern similar research themes, i.e., spatial representation, real-time planning, perception, advanced computer architectures, etc.

II. MEETING LOGISTICS:

The DOE/CESAR meeting will be held at UCLA on August 18-19, 1985 in conjunction with the forthcoming International Joint Conference on Artificial Intelligence. The CESAR Program is prepared to pay for all incremental costs (e.g., price of additional hotel fee, etc.) for workshop participants who will be attending the IJCAI meeting, additional transportation costs for those who do not plan to participate in IJACI. Based upon response to this (and related) initial solicitations, formal invitations will be extended by CESAR/ORNL in May, and a tentative agenda will be distributed in June.

III. MEETING FORMAT:

We currently anticipate attendance of ~25, with invitations extended to technical leaders of advanced programs related to planning and sensing by intelligent machines. Initial informal contacts have been made with individuals at SRI, FMC, NOSC, CMU, etc. The first morning (August 18) will have overview presentations; the remainder of the workshop will involve free wheeling discussions and panels.

IV: <u>ACTION ITEMS:</u>

If you would be interested in participating in the DOE/CESAR Workshop please complete the following form and return it to:

Dr. C. R. Weisbin, Director
Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
Building 6025 - Room 6N
P.O. Box X
Oak Ridge, TN 37831

NAME: ______________________________
(PLEASE PRINT OR TYPE)

ORGANIZATION: ______________________________

____

ADDRESS: ______________________________

____

TELEPHONE NO. ______________________________

TECHNICAL RESEARCH INTERESTS: ______________________________

____

I (would/would not) be able to present an overview on

the subject of: ______________________________

____

cc: F. C. Maienschein
O. P. Manley, DOE/HQ
P. Cheeseman, SRI
J. Crowley, CMU
K. N. Reid, OSU

AGENDA

AGENDA

CENTER FOR ENGINEERING SYSTEMS ADVANCED RESEARCH WORKSHOP

August 18 and 19, 1985

Sunday, August 18

8:00 - 10:15	First three overview papers and plenary discussion
10:15 - 10:30	Coffee Break
10:30 - 12:00	Last two overview papers
12:00 - 1:30	Lunch and informal discussion
1:30	Break up into small subgroups to prepare draft answers to workshop themes
3:00	Coffee Break
5:30	Draft responses to C. R. Weisbin

Monday, August 19

8:00 - 10:15	Draft responses presented in plenary session for feedback by entire group
10:15 - 10:30	Coffee Break
10:30 - 12:00	Draft responses continued
12:00 - 1:30	Lunch and informal discussion
1:30 - 3:00	Break up into small subgroups to revise draft based on morning discussion
3:00 - 3:15	Coffee Break
3:15 - 6:00	Presentation of final responses to workshop themes

ADJOURN

FULL PAPERS

FULL PAPERS

Paper 1

Multiple Levels of Representation and Problem-Solving Using Maps From Sonar Data

Alberto Elfes

The Robotics Institute
Carnegie-Mellon University
Pittsburgh, PA 15213

Abstract

This paper describes a sonar-based mapping and navigation system for autonomous mobile robots operating in unknown and unstructured surroundings. The system uses sonar range data to build a multi-leveled and multi-faceted description of the robot's operating environment. Sonar maps are represented in the system along several dimensions: the Abstraction *axis, the* Geographical *axis, and the* Resolution *axis. Different kinds of problem-solving activities can be performed and different levels of performance can be achieved by working with these multiple representations of maps. The major modules of the* Dolphin *system are described and related to the various mapping representations used. The system is also situated within the wider context of developing an advanced software architecture for autonomous mobile robots.*

1. Introduction

The Dolphin system is intended to provide sonar-based mapping and navigation for an autonomous mobile robot operating in unknown and unstructured environments. The system is completely autonomous in the sense that it has no *a priori* knowledge of its surroundings and also carries no user-provided map data. It acquires data from the real world through a set of sonar sensors and uses the interpreted data to build a multi-leveled and multi-faceted description of the robot's operating environment. In *Cruising* mode, the system acquires data, builds maps, plans safe paths and navigates towards a given goal. In *Exploration* mode, it can wander around and collect enough information so as to be able to build a good description of its environment.

The system is intended for indoor as well as outdoor use; outdoors, it may be coupled to other systems, such as vision, to locate landmarks that would serve as intermediate or final destinations.

In the course of this paper, we will briefly describe a general framework for mobile robot software, situate the present system within this framework, discuss the multiple representations used for sonar maps as well as their use in different kinds of problem-solving activities, and conclude with a description of the overall system architecture.

2. A Conceptual Framework for Autonomous Mobile Robot Software

Research in mobile autonomous vehicles provides a very rich environment for the development and test of advanced concepts in a variety of areas, such as Robotics, Artificial Intelligence, Sensor Understanding and Integration, Real-World Modelling, Planning and Control. Some research efforts, however, have tended to address only very specific problems in robotics and mobility, while ignoring more global issues.

A premise of the work described in this paper is that the levels of autonomy and performance essential for a mobile robot will emerge on one hand through research in specific problem areas, but also, on the other hand, by investigating the integration and coupling of individual problem-solving elements (such as a path-planning module or a sonar mapping algorithm) into a cooperating whole.

In our research on the concepts and tools necessary for the development of a general architecture for autonomous mobile robot software, we identified seven conceptual levels of activities that are needed in a mobile system (Fig. 2-1):

- *Robot Interface*: This level takes care of the physical control of the different sensors and actuators available to the robot. It provides a set of well-defined primitives for locomotion, sensor control, data acquisition, etc. that serve as an interface, allowing the higher levels of the system to be programmed "device-independently". It includes activities such as actuator control by Actuator Modules, and dead-reckoning estimation of robot position and orientation. *Internal Sensors* provide information on the status of the different physical subsystems of the robot, while *External Sensors* are used to acquire data from the robot's environment.

- *Sensor Interpretation*: On this level the acquisition of sensor data and its interpretation by Sensor Modules is done. Each Sensor Module is specialized in one type of sensor or even in extracting a specific kind of information from the sensor data. They provide information to the higher levels using a common representation and a common frame of reference.

- *Sensor Integration*: Here the integration of information coming from qualitatively different sensors is performed. This is done by taking pieces of interpreted data provided by the Sensor Modules and correlating them to each other. For example, geometric boundaries of an obstacle

VII. Global Control

- Global Control of System Behaviour
- Scheduling of Activities
- Integration of Plan-Driven with Data-Driven Activities

VI. Global Planning

- Task-Level Planning to provide sequences of sensory, actuator and problem-solving (software) actions
- Simulation
- Error-Recovery and Replanning in case of failure or unexpected events

V. Problem-Solving

- Problem-Solving Modules provide services such as Path-Planning, Obstacle Avoidance, Internal Sensor Monitoring, User Interface, etc.

IV. Real-World Modelling

- Integration of local pieces of correlated information into a Global Real-World Model that describes the robot's environment of operation
- Matching acquired information against stored maps
- Object Identification
- Landmark Recognition

III. Sensor Integration

- Information provided by different Sensor Modules is correlated and abstracted
- Common representations and compatible frames of reference are used

II. Sensor Interpretation

- Acquisition of Sensor Data (Vision, Sonar, Rangefinder, etc.)
- Interpretation of Sensor Data

I. Robot Interface

- Set of Primitives for Robot Operation
- Actuator Control (e.g., locomotion)
- Sensor Control

Figure 2-1: Conceptual Activity Levels for a Mobile Robot Software Architecture.

extracted by sonar can be projected onto an image provided by the vision subsystem and can help in identifying a certain object. On this level, information is aggregated and assertions about specific parts of the real world can be made.

- *Real-World Modelling*: Partial, aggregated and local pieces of information are used in the incremental construction of a coherent Global Real-World Model of the robot's environment; this Model can then be used for several other activities, such as landmark recognition, matching of newly acquired information against already stored maps, and generation of expectancies and goals.

- *Problem-Solving*: In the context of autonomous locomotion, a variety of problem-solving activities are necessary, such as path-planning, monitoring of internal sensors, obstacle-avoidance, interfacing to a human user, etc. These different activities are performed by Server Modules that provide specific services.

- *Global Planning*: This level provides task-level planning for autonomous generation of sequences of actuator, sensor and processing operations to achieve a global goal proposed to the robot. Other necessary activities include simulation, error detection, diagnosis and recovery, and replanning in the case of unexpected situations or failures.

- *Global Control*: Finally, on this level Supervisory Modules are responsible for the scheduling of different activities and for combining Plan-driven with Data-driven activities in an integrated manner so as to achieve coherent behaviour.

Clearly, none of the presently existing mobile robot systems covers all of the levels described above. This conceptual structure provides, however, a context within which several of our research efforts situate themselves [2, 5, 6]. The Dolphin system for sonar-based mapping and navigation, in particular, embodies several of the elements of the framework, as discussed in Section 5.

3. Building Sonar Maps

3.1. Introduction

Several of the efforts towards autonomous navigation in unstructured environments have used stereo vision to extract 3D information from the robot's surroundings [4, 3, 10]. One of the major difficulties with this approach is that the resulting maps are typically very sparse, due to the intrinsic computational expense of extracting range data from stereo pairs of images. This limitation led us to explore the use of an alternative kind of sensor, such as sonar, that could deliver range information directly.

The Dolphin sonar system is able to build dense maps of the robot's environment and classify regions as EMPTY, OCCUPIED or UNKNOWN. The central representation of sonar mapping information is called the *Probabilistic* or *Sensor-Level Local Map*, which uses a medium-resolution grid (typically 0.5 ft). Information about empty, occupied and unknown areas, as well as the associated confidence factors, is stored in the cells of a two-dimensional array. These sonar maps are very useful for motion planning. They are much denser than those made by the stereo vision programs, and computationally at least one order of magnitude faster to produce.

Presently, the cycle of operation of the sonar system is as follows: from its current position, the system acquires a set of range measurements provided by the sonar sensors; these readings are then interpreted as providing assertions concerning *empty* and *occupied* areas, and serve to update the sonar map. The map is now used to plan safe paths around obstacles, and the robot moves a certain distance along the path. It updates its position and orientation estimate and repeats the cycle.

3.2. The Sonar Sensor Subsystem

The sonar devices chosen are Polaroid laboratory grade ultrasonic transducers [8]. These sonar elements have a useful measuring range of 0.9 to 35.0 ft, with an accuracy on the order of 1 %. The main lobe of the sensitivity function corresponds to a beam angle of 30° at -38 dB. The system is optimized for giving the range of the nearest sound reflector in its field of view, and works well for this purpose.

The sonar sensory system was built at Denning Mobile Robotics, and was mounted on two different robots (*Neptune* [7] for indoor use, and the *Terragator* for outdoors). It is composed of a ring of 24 Polaroid sensors spaced 15° apart, and a Z80 controlling microprocessor that selects and fires the sensors, timing the returns and providing range values. This range information is then sent over a serial link to a VAX mainframe, where the interpretation of the sonar data and the higher level mapping and navigation functions are presently performed.

3.3. Approach

In this section we will briefly review the Local Map building process (described in detail in [5]), and in the next section we will discuss how other representations are derived from it.

There are a number of problems inherent to the data obtained from the sonar device: the timing circuitry causes imprecision in the distance measured; multiple reflections or reflections away from the sensor, due to a low angle of incidence on a specular surface, generate erroneous readings; finally, the wide angle of the sonar beam imposes only a very loose constraint on the position of the detected object.

These conditions led us to consider a probabilistic approach to the interpretation of range data and the building of sonar-based maps.

Our method starts with a number of range measurements obtained from Polaroid sonar units whose position with respect to the robot is known. Each measurement provides information about *probably empty* and *possibly occupied* volumes in the space subtended by the beam (a 30^{0} cone for the present sensors). This occupancy information is projected onto a rasterized two-dimensional horizontal map. Sets of readings taken both from different sensors and from different positions of the robot are progressively incorporated into the sonar map. Primarily because of the wide beam angle, the sonar data provides only indirect information about the location of the detected objects. The constraints obtained from individual readings are combined to reduce the uncertainty. As more readings are added the area deduced to be empty expands, and the expanding empty area encroaches on and sharpens the possibly occupied region. The map becomes gradually more detailed.

The sonar beam is modelled by probability distribution functions. Informally, these functions describe our confidence that the points inside the cone of the beam are empty and our uncertainty about the location of the point that caused the echo. The functions are based on the range value and on the spatial sensitivity pattern of the sonar.

3.4. Representing Maps

Local Sonar Maps are two-dimensional arrays of cells corresponding to a horizontal grid imposed on the area to be mapped. The grid has $M \times N$ cells, each of size $\Delta \times \Delta$. In each cell we store information that describes its status (UNKNOWN, EMPTY or OCCUPIED) and the associated certainty factors. The following convention is used to represent map information:

UNKNOWN	0
EMPTY	$[-1, 0)$
OCCUPIED	$(0, 1]$

A cell is considered UNKNOWN if no information concerning it is available. A cell (x_i,y_j) can be EMPTY with a confidence factor $Emp(x_i,y_j)$ (corresponding to values from 0 to -1) and OCCUPIED with a degree of certainty $Occ(x_i,y_j)$ ranging from 0 to 1. Due to sonar and reflection errors, we may have conflicting information in a given cell (x_i,y_j). A measure of this disparity is given by: $Error(x_i,y_j) = 1 - (|Emp(x_i,y_j)| + |Occ(x_i,y_j)|)$.

3.5. Composing Information from Several Readings

To build a Sonar Map, we compute the *empty* and *occupied* sonar beam probability distributions for individual range readings, then combine them with the information already stored in the map. The position and orientation of the sonar sensor is used to register the beam with the map, and the beam probabilities are then projected onto the discrete map cells.

Each sonar reading provides partial evidence about a map cell being OCCUPIED or EMPTY. This evidence is combined with existing data to refine the status of each cell. The *evidence combination rules* that control this process allow the new evidence to enhance or weaken existing hypotheses. Different readings asserting that a cell is EMPTY will enhance each other, as will readings implying that the cell is OCCUPIED; on the other hand, evidence that the cell is EMPTY will weaken the certainty of it being OCCUPIED and vice-versa.

One range measurement contains only a small amount of information. By combining the evidence from many readings as the robot moves in its environment, the area known to be empty is expanded. The number of regions somewhere containing an occupied cell increases, while the range of uncertainty in each such region decreases. The overall effect as more readings are added is a gradually increasing coverage along with an increasing precision in the object locations. Typically after a few hundred readings (and less than a second of computer time) our method is able to "condense out" a comprehensive map covering a thousand square feet with better than one foot accuracy in the position of the objects detected.

3.6. Maps

A typical map obtained through the method outlined above is shown in Fig. 3-1, and the corresponding certainty factor distributions are shown in Figs. 3-2 and 3-3. These are the maps obtained after doing a thresholding step, where OCCUPIED and EMPTY values are compared and a final decision is made concerning what label to attach to each cell.

4. Multiple Axis of Representation of Sonar Mapping Information

From the Probabilistic Local Maps described in the previous section, several other data structures are derived. We use the following dimensions of representation (Fig. 4-1):

- THE ABSTRACTION AXIS: Along this axis we move from a sensor-based, data-intensive representation to increasingly higher levels of interpretation and abstraction. Three levels are defined: the *Sensor Level*, the *Geometric Level* and the *Symbolic Level*.

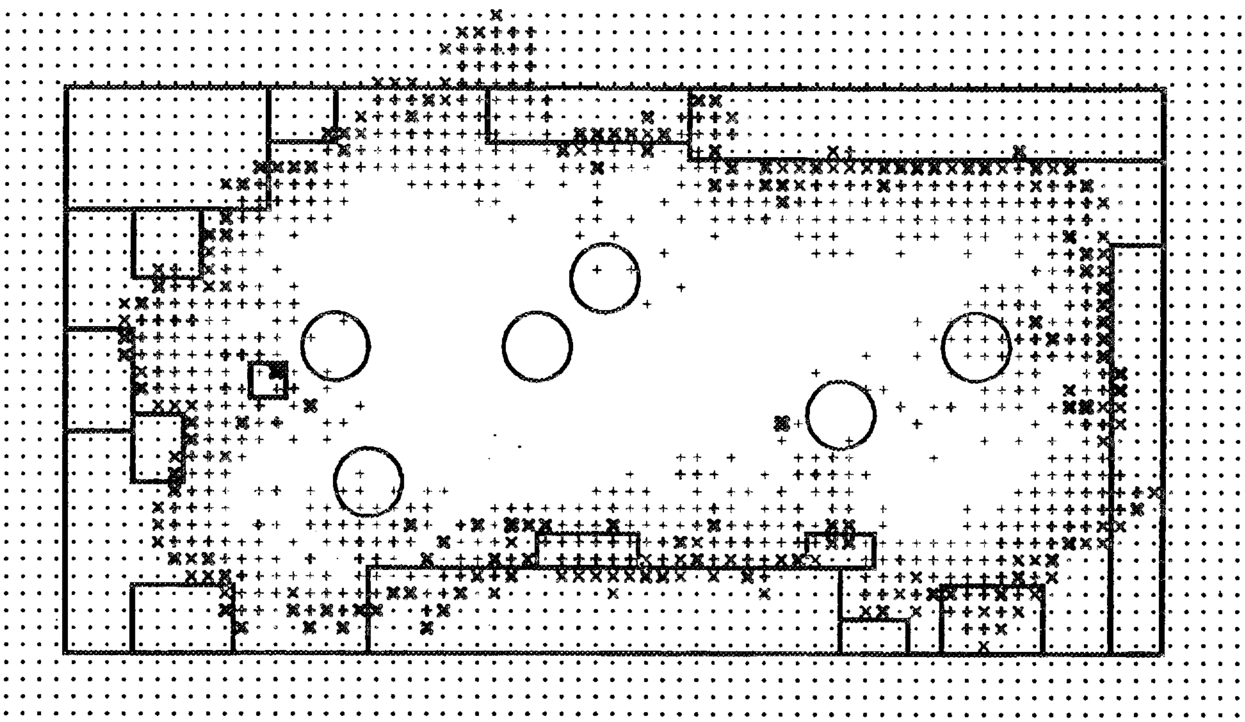

Figure 3-1: A Two-Dimensional Sonar Map. *Empty* areas with a high certainty factor are represented by white areas; lower certainty factors by " + " symbols of increasing thickness. *Occupied* areas are represented by "×" symbols, and *Unknown* areas by "·" . The position of the robot is shown by a circle and the outline of the room and of the major objects by a solid line.

- THE GEOGRAPHICAL AXIS: Along this axis we define *Views*, *Local Maps* and *Global Maps*, depending on the extent and characteristics of the area covered.

- THE RESOLUTION AXIS: Sonar Maps are generated at different values of grid resolution for different applications. Some computations can be performed satisfactorily at low levels of detail, while others need high or even multiple degrees of resolution.

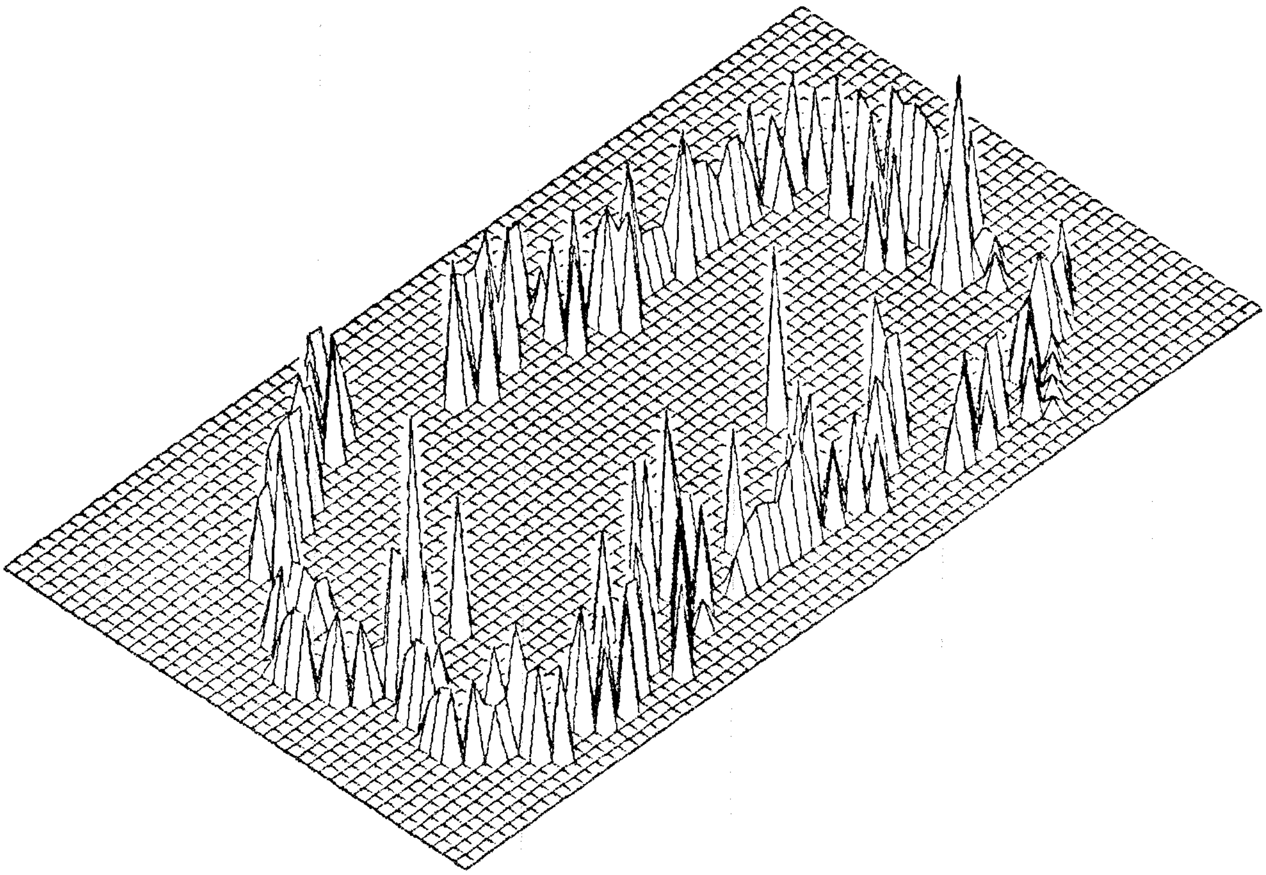

Figure 3-2: The Occupied Areas in the Sonar Map. This 3-D view shows the Certainty Factors $Occ(x_i, y_j)$.

4.1. The Abstraction Axis

The first kind of sonar map built from the sonar range readings uses the *Probabilistic* representation described earlier. A two-dimensional grid covering a limited area of interest is used. This map is derived directly from the interpretation of the sensor readings and is, in a sense, the description closest to the real world. It serves as the basis from which other kinds of representation are derived. Along the Abstraction Axis, this data-intensive representation is also defined as the *Sensor Level* map.

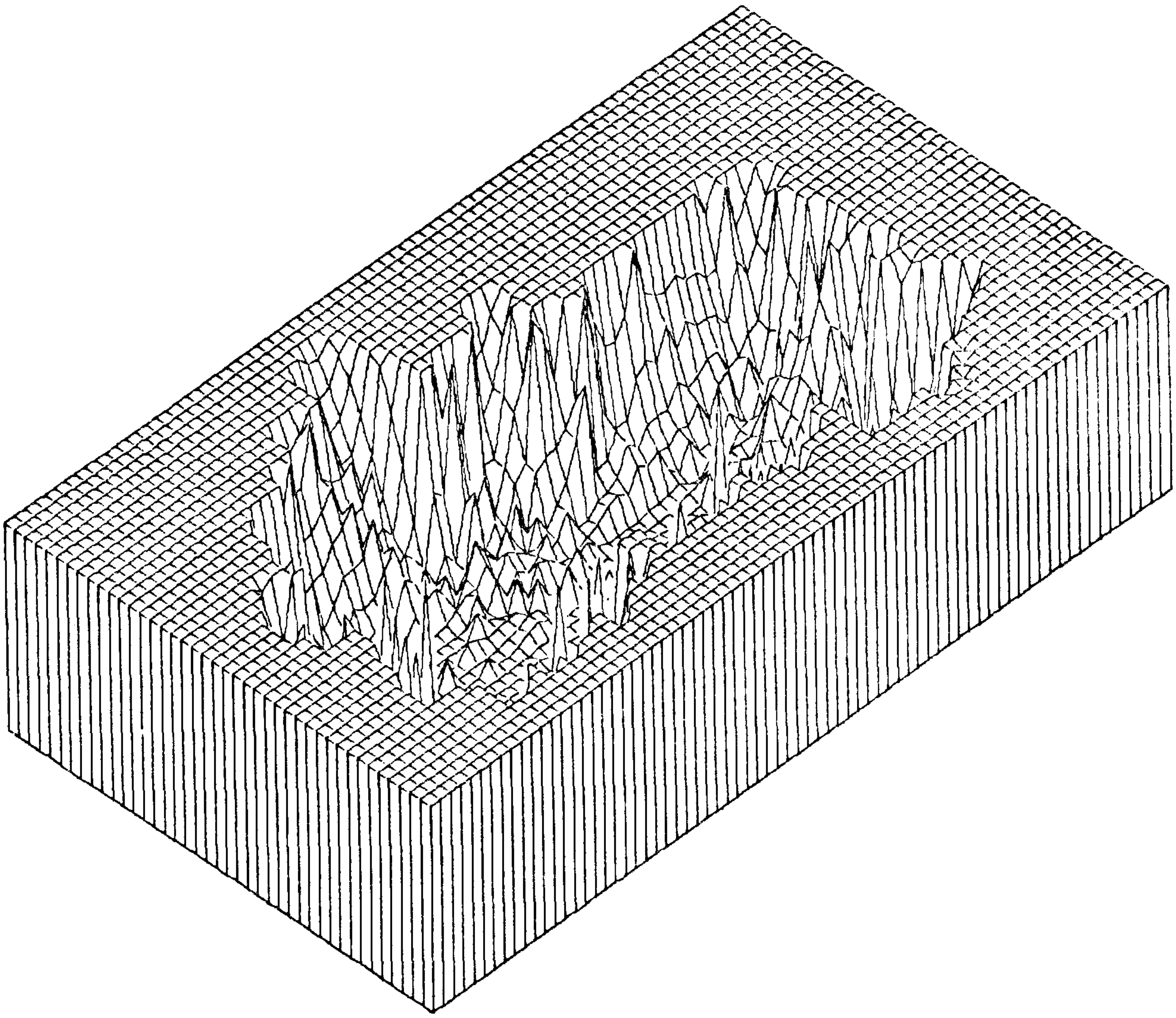

Figure 3-3: The Empty Areas in the Sonar Map. This 3-D view shows the Certainty Factors $Emp(x_i, y_j)$.

The second level is called the *Geometric Level.* It is built by scanning the Sensor Level Map and identifying blobs of cells with high OCCUPIED confidence factors. These are merged into uniquely labeled objects with explicitly represented polygonal boundaries. If needed, the same can be done with EMPTY areas.

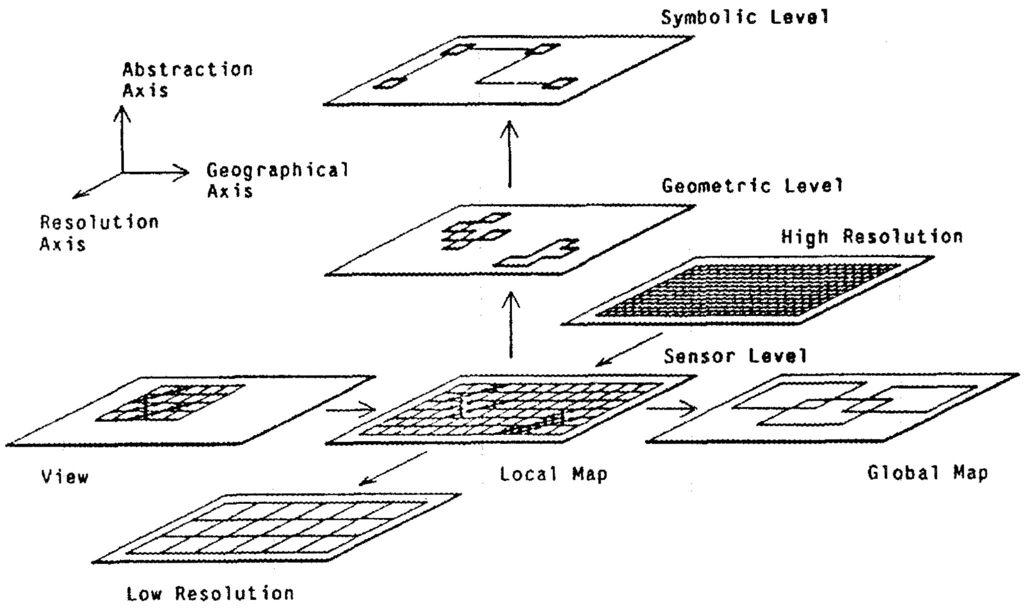

Figure 4.1: Multiple Axis of Representation of Sonar Maps.

The third is the *Symbolic Level*, where maps of larger areas (typically Global Maps) are described using a graph-like representation. This description bears only a topological equivalence to the real world. Nodes represent "interesting" areas, where more detailed mapping information is necessary or available, while edges correspond to simpler or "uninteresting" (navigationally speaking) areas, such as corridors.

Different kinds of problem-solving activities are better performed on different levels of abstraction. For example, global path-planning (such as how to get from one building wing to another) is done on the symbolic level, while navigation through a specific office or lab is done on the sensor-level map, where all the detailed information about objects and free space, as well as the associated certainty factors, is stored.

4.2. The Geographical Axis

In order to be able to focus on specific geographic areas and to handle portions of as well as complete maps, we define a hierarchy of maps with increasing degrees of coverage. Progressing along the Geographical Axis, we start with *Views*, which are maps generated from scans taken from the current position, and that describe the area visible to the robot from that place. As the vehicle moves, several Views are acquired and integrated into a *Local Map*. The latter corresponds to physically delimited spaces such as labs or offices, which define a connected region of visibility. *Global Maps* are sets of several Local Maps, and cover wider spaces such as a whole wing of a building, with labs, offices, open areas, corridors, etc.

4.3. The Resolution Axis

Finally, along the Resolution Axis, we again start with the Local Probability Maps and generate a progression of them, with increasingly less detail. This permits certains kinds of computations to be performed either at lower levels of resolution with correspondingly less computational expense, or else allows operations at coarser levels to guide the problem-solving activities at finer levels of resolution.

The finest sonar maps that can be obtained from the method outlined in Section 3 (considering the limitations intrinsic to the sensor) have a cell size of 0.1×0.1 ft . For navigation purposes, we have typically been using a 0.5 ft grid for indoors and a 1.0 ft grid for outdoors. Nevertheless, several operations on the maps are expensive and are done more quickly at even lower levels of resolution. For these cases we reduce higher resolution maps by an averaging process that produces a coarser description. One example of an application of this technique is the Map Matching procedure described in [5]: two Local Maps being compared with each other are first matched at a low level of detail; the result then constrains the search for a match at the next higher level of resolution.

5. Overall System Architecture

To provide a context for these multiple descriptions, we will briefly present the current architecture of Dolphin, and show how the different dimensions and levels of representation of sonar maps interact with and are used by the various problem-solving activities that happen in the system. We will also situate it within a more global architecture, and discuss its relationship to the conceptual framework outlined in Section 2.

The overall architecture of the Sonar Mapping and Navigation part of the Dolphin system is shown in Fig. 5-1. The function of the major modules and their interaction with the different sonar map representations is described below:

Sonar Control: Interfaces to and controls the Sonar Sensor Ring, providing range readings.

Scanner: Preprocesses and filters the sonar data. Annotates it with the position and orientation of the sensor.

Mapper: Using the information provided by the Scanner, generates a View obtained from the current position of the robot. This View is then integrated into a Local Map.

Cartographer: Aggregates sets of Local Maps into Global Maps. Provides map bookkeeping functions.

Matcher: Matches a newly acquired Local Map against portions of Global Maps for operations such as landmark identification or providing an alternative update for the global (absolute) robot position and orientation estimate.

Object Extraction:
Identifies obstacles by merging blobs of OCCUPIED cells and extracting the corresponding polygonal boundaries.

Graph Building: Searches for regions with simple or complex patterns of obstacles to identify "interesting" and "free" spaces.

Path-Planning: Three levels of path-planning are possible: *Symbolic Path-Planning* is done over wider areas (Global Maps) and at a higher level of abstraction (Symbolic Maps); *Geometric Path-Planning* is done as an intermediary stage, when the uncertainty in Local Maps is low; and *Sensor Map Path-Planning* is used to generate safe paths, taking into account the certainty factors. The path generated is provided to the Navigator.

Navigator: Takes care of the overall locomotion control of the vehicle. This includes examining already planned paths to determine whether they are still usable, invoking the path-planners to provide new paths, overseeing the actual locomotion, setting intermediary goals, etc.

Conductor: Controls the physical locomotion of the robot vehicle along the proposed path. Provides an estimate of the new position and orientation of the robot.

Guardian: During actual locomotion, this module checks the incoming sonar readings and signals a stop if the robot is coming too close to a (possibly moving) obstacle not detected previously. It serves as a "sonar bumper".

Supervisor: Takes care of the overall control of the system.

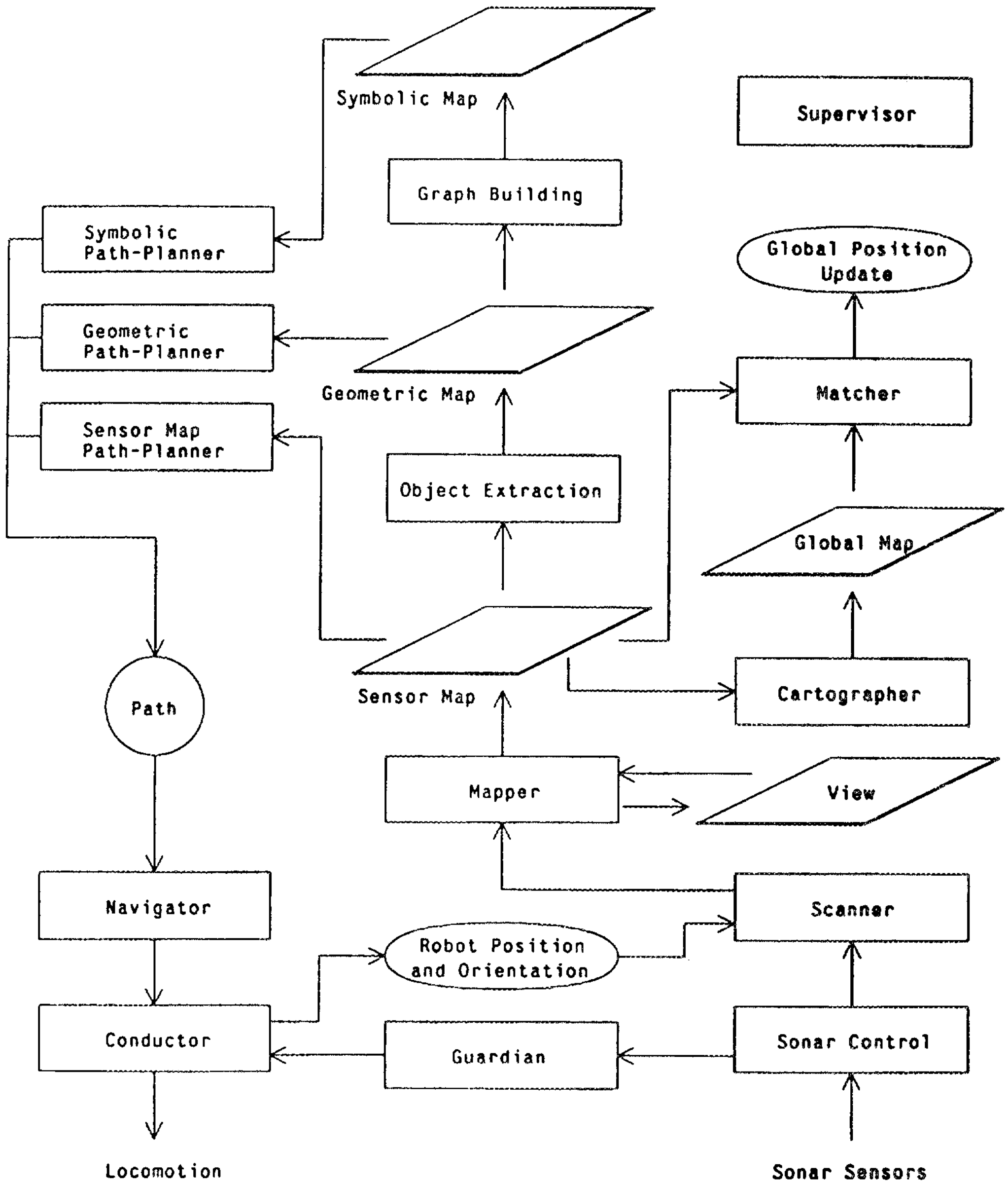

Figure 5.1: Architecture of the Sonar Mapping and Navigation System

Comparing this architecture with the conceptual framework outlined in Section 2, we can identify the following correspondence: the Sonar Control and Conductor modules belong to Level I; Scanning, Mapping, Object Extraction and Graph Building provide functions on Level II; the Cartographer and the Matcher operate on Level IV; Path-Planning, Navigation and the Guardian are situated in Level V; and the Supervisor is in Level VII.

5.1. Extending the Architecture

The implementation described above embodies a sequential control-flow organization. This, however, does not reflect the intrinsic problem-solving characteristics inherent to mobile robot software. The various modules involved in the problem-solving effort are frequently quasi-independent and have a low degree of coupling; therefore, they should conceptually proceed in parallel, interacting with each other as needed.

It is in this context that we designed a Distributed Problem-Solving framework within which the kinds of parallel and coordinated activities needed for a mobile robot could be expressed naturally [2]. This framework offers parallelism on the *process* level. Conceptually, it provides a computing environment where the problem-solving activities are performed by several independent processes. These can communicate with each other through messages as well as post on or retrieve relevant information from multiple Blackboards. A set of primitives was implemented that provide message-based communication, process control, blackboard creation and access, and event handling [1].

This framework was used to design a Distributed Control System to supervise and coordinate the activities of a mobile robot [2]. The different tasks are handled by independent Expert Modules; each is a pair of <master, slave> processes, where the master controls the scheduling and the activities of the slave. Communication among Expert Modules occurs asynchronously over a Blackboard structure encapsulated in a Blackboard Monitor. Information specific to the accomplishment of an overall goal is provided through a Control Plan. The system can be distributed over a network of processors; an Executive local to each processor and an interprocess message communication mechanism ensure transparency of the underlying network structure.

We have recently started the implementation of a distributed version of Dolphin as an actual testbed for these ideas [6]. This Control System would correspond to a Level VII activity.

Moving towards a higher degree of autonomy and flexibility, we are also beginning to address the development of a task-level Global Planner that would automatically generate the Control Plan mentioned above. We are considering a hierarchical approach similar to NOAH [9], using a graph to represent the plan and explicitly storing alternatives and sensor-dependent conditions as part of it. The elementary operations of sensor information gathering, interpretation, actuator control and specific problem-solving activities are the primitives on which the planner bases its plan.

A simplified view of an expanded version of the Dolphin system, including Distributed Control and a Global Planner, is shown in Fig. 5-2. The Control Blackboard stores the more relevant pieces of high-level information needed for overall coordinated behaviour. Complex sub-systems such as sensor processing may have independent blackboards of their own.

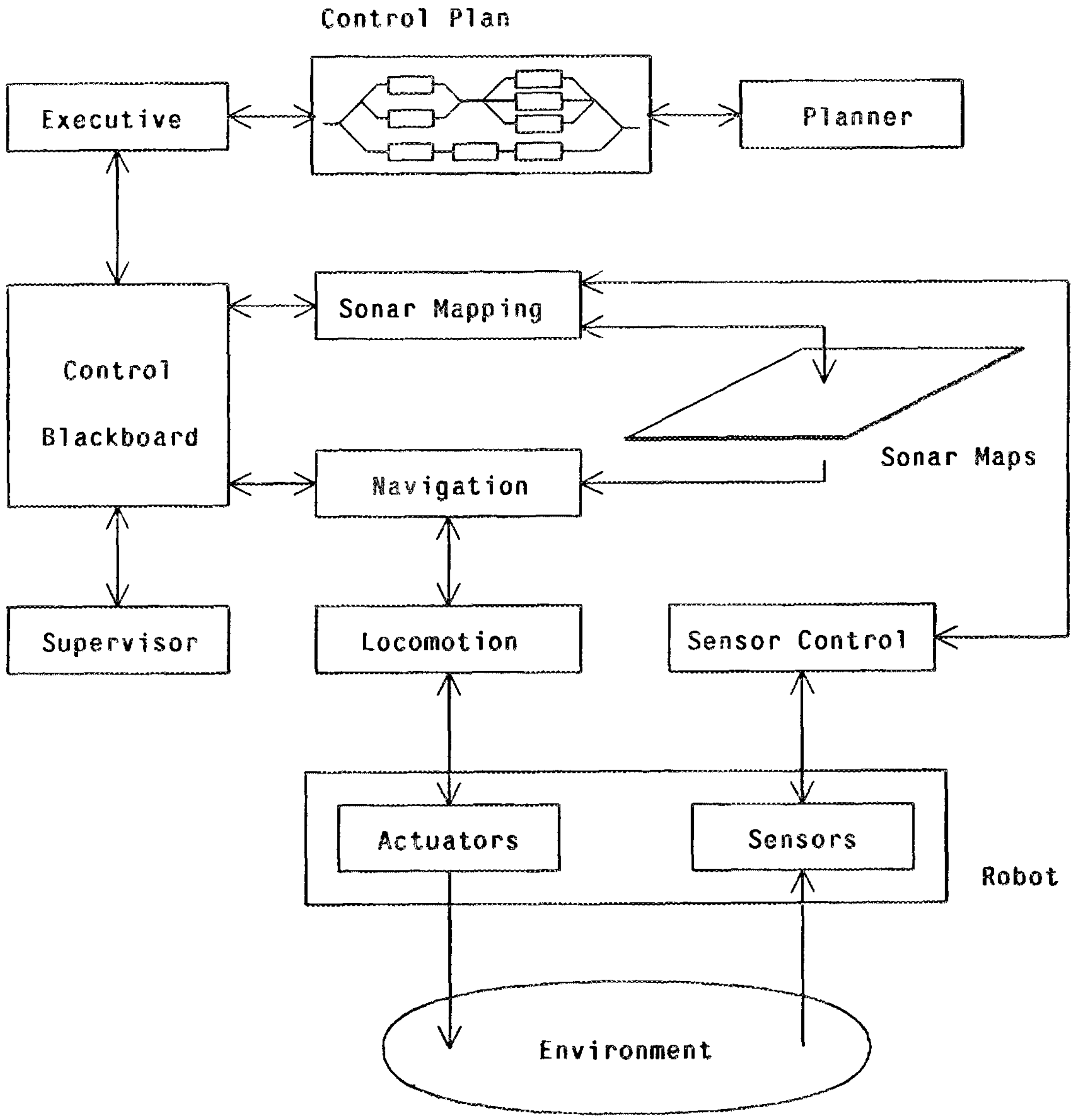

Figure 5-2: General Architecture of the Dolphin System

6. Tests of the System

The Dolphin sonar-based mapping and navigation system described here was tested in several indoor runs in cluttered environments using the *Neptune* mobile robot [7], developed at the Mobile Robot Laboratory of the Robotics Institute, CMU. It was also tested in outdoor environments, operating among trees, using the *Terragator* robot, developed at the Robotics Construction Laboraty, CMU. The system operated successfully in both kinds of environments, navigating the robot towards a given destination.

7. Conclusions

We have described a system that uses a Sensor Level, probability-based sonar map representation of medium resolution to build several kinds of maps. Three different dimensions of representation are defined: the Abstraction Axis, the Geographical Axis and the Resolution Axis. These maps are used by a sonar mapping and navigation system that performed successfully in indoor and outdoor environments. We are now expanding the system to test distributed control and global planning mechanisms.

8. Acknowledgments

The author would like to thank Hans P. Moravec for his support and for providing several important insights into the topics discussed in this paper, Gregg W. Podnar for providing assistance with the Neptune robot, Michael Fuhrman for advice on the use of an obscure 3D drawing package, and Richard Redpath for providing more reliable communication to the sonar ring, as well as for assistance during outdoor runs.

This research is being supported in part by the Office of Naval Research under Contract N00014-81-K-0503, in part by Denning Mobile Robotics, Inc., and in part by the Western Pennsylvania Advanced Technology Center. The author is supported in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq, Brazil, under Grant 200.986-80, in part by the Instituto Tecnológico de Aeronáutica - ITA, Brazil, and in part by The Robotics Institute, Carnegie-Mellon University.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

References

[1] Cardozo, E. and Elfes, A.
A Set of Primitives for Distributed Problem-Solving.
Technical Report, The Design Research Center, Carnegie-Mellon University, 1985.
In preparation.

[2] Elfes, A. and Talukdar, S.N.
A Distributed Control System for the CMU Rover.
In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence - IJCAI-83.* IJCAI, Karlsruhe,Germany, August, 1983.

[3] Matthies, L.H. and Thorpe, C.E.
Experience With Visual Robot Navigation.
In *Proceedings of IEEE Oceans 84.* IEEE , Washington, D.C., August, 1984.

[4] Moravec, H.P.
Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover.
PhD thesis, Stanford University, September, 1980.
Also available as Stanford AIM-340, CS-80-813 and CMU-RI-TR-01-82, 1982; and published as *Robot Rover Visual Navigation* by UMI Research Press, Ann Arbor, Michigan, 1981.

[5] Moravec, H.P. and Elfes, A.
High-Resolution Maps from Wide-Angle Sonar.
In *IEEE International Conference on Robotics and Automation.* IEEE, March, 1985.

[6] Moravec, H.P. et al.
Towards Autonomous Vehicles.
1985 Robotics Research Review.
The Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA, 1985.

[7] Podnar, G.W., Blackwell, M.K. and Dowling, K.
A Functional Vehicle for Autonomous Mobile Robot Research.
Technical Report CMU-RI-TR-84-28, The Robotics Institute, Carnegie-Mellon University, April, 1984.

[8] Polaroid Corporation.
Ultrasonic Range Finders.
Polaroid Corporation, 1982.

[9] Sacerdoti, E.D.
A Structure for Plans and Behavior.
Elsevier, New York, N.Y., 1977.

[10] Thorpe, C.E.
The CMU Rover and the FIDO Vision and Navigation System.
In *Symposium on Autonomous Underwater Robots.* University of New Hampshire, Marine Systems Engineering Lab, May, 1983.

Paper 2

The Synthesis of Digital Machines with Provable Epistemic Properties

Stanley J. Rosenschein
Leslie Pack Kaelbling

Artificial Intelligence Center
SRI International
and
Center for the Study of Language and Information
Stanford University

September 18, 1985

Abstract

Many artificial intelligence applications involve the design of systems intended to track and react to conditions in their physical environments in real time. Real-time performance is difficult to achieve using traditional AI techniques because of their reliance on expensive runtime symbolic inference. This paper addresses this problem by describing a mathematical framework and design tools for analyzing and synthesizing machines with compiled knowledge. The concept of knowledge is formulated math-matically in terms of the relationship between states of a machine and states of its environment over time. The design approach is based on the use of metaprograms to compute a machine description which can then be transformed either into physical circuitry or into code that simulates the described machine. The compilation of knowledge is facilitated by parameterizing machine constructors by other machine constructors and by objects usually encoded as runtime structures.

1 Introduction

Many important computer applications involve the design of hardware and software that are part of a larger system embedded in a physical environment. Applications of this kind arise in process control, avionics, robotics, and artificial intelligence; in the typical case, the computer's principal task is to track and react to conditions in the environment. For the system to operate as desired, it must be designed to recognize the relevant environmental conditions and to compute appropriate responses when required. As more open-ended environments are considered and as the conditions to be recognized and the responses to be supplied become more complex, the job of designing real-time embedded systems becomes correspondingly more difficult.

The problem is particularly acute in the design of highly reactive AI systems, such as intelligent robots. A robot can be viewed abstractly as a complex control system that monitors sensory inputs and acts to achieve or maintain certain goal conditions in its environment. In simple control systems, facts about the environment can often be encoded as a small set of numerical parameters. More complex kinds of information, however, such as those needed by intelligent robots, require correspondingly more complex data structures for their encoding. Moreover, real-time performance requires that there be a constant bound on the number of computational operations performed between inputs and outputs.

The Artificial Intelligence Center at SRI International is designing and implementing a mobile robot in the tradition of Shakey [13]. The aim of this project is to combine significant perceptual, reasoning, and communication abilities in an autonomous computer-controlled device and to have it operate in real time. In attempting to reconcile the goal of manipulating complex information with that of real-time operation, we have adopted a design approach based on (1) the use of epistemic logic in the formal analysis of the robot's information states and (2) the use of metaprograms that automatically construct real-time control programs amenable to this type of formal analysis.

Real-time performance is difficult to achieve using traditional AI techniques. This difficulty stems, in part, from a failure to distinguish between two types of facts that are relevant to a robot's operation. The first of these can be called the *dynamic* facts, as they involve moment-to-moment conditions of the environment. The second type comprises the *permanent* or *static* facts, i.e. those which are perhaps better thought of as part of a model of the environment in which the robot operates. The traditional AI approach to the encoding of information ("knowledge representation") is to think about *all* these facts as objects of the same sort and to encode them uniformly as symbolic data structures that are manipulated by the program. This approach is attractive because it seems to offer the possibility of reducing the problem of designing intelligent machines to the conceptually simpler task of constructing programs that syntactically derive consequences of facts in a knowledge base [11,10].

As attractive as this strategy may be, its implementation raises serious technical difficulties which derive from the computational complexity of inference. It is well recognized that the more open-ended the environment, the more expressive the logic needed to describe it and the less tractable is the problem of reasoning explicitly in the logic. In some applications, the moment-to-moment synchronization of the programs with conditions in the surrounding world can be conveniently ignored. In such domains (e.g. theorem proving, medical diagnosis, geology, and organizational behavior), the time complexity of inference is not a critical problem; thus, the implementation of intelligent information processing by means of conventional symbolic inference techniques is feasible.

However, in the mobile-robot domain, the permanent facts relevant to time-critical, low-level interpretation and decision-making are so complex that is it impossible to reason with them explicitly in real time. This point is hardly controversial; the assumption is generally made that in applications of this sort, static knowledge must be "compiled in." This paper explores the idea of knowledge compilation from a theoretical standpoint and suggests how it might be applied at various levels in the construction of intelligent systems, thus avoiding certain problematic aspects of general deductive inference.

Much work on formalizing properties of knowledge has been done in philosophy [5,9], theoretical computer science [3], and AI [12,10,8]. Most of the work in this tradition is carried out in an abstract setting; the essential concept of *knowledge* is not given a concrete physical or computational interpretation. Where such an interpretation *is* given, it is usually in terms of procedures that manipulate sentences of a formal language, often ignoring issues of computational complexity. The situated-automata approach attempts to avoid inferential complexity by providing a concrete computational model for epistemic logic in a framework that does not depend on viewing the system as manipulating sentences of a logic [14].

In the situated-automata framework, the concept of knowledge is analyzed in terms of logical relationships between the state of a process (e.g., a machine) and that of its surrounding world. Because of constraints between a process and its environment, not every state of the machine-environment pair is possible, in general. A process π is said to know a proposition ϕ in a situation where its internal state is s, if in all possible situations in which π is in state s, ϕ is satisfied. This definition of knowledge satisfies the axioms of modal system S5, including deductive closure and positive and negative introspection.

In its original formulation, situated-automata theory dealt with the state of a system as an unanalyzed whole. Since machines designed for real applications can take on an enormous number of states, they must be built hierarchically, with the size of the state set growing as the product of the sizes of the state sets of the component machines. This paper extends situated-automata theory to hierarchically constructed machines in order to facilitate the epistemic analysis of composite machines.

On the practical level, this approach has led to the development of Rex, a set of tools for constructing complex programs with rigorously definable epistemic properties. Instead of constructing a description of the target machine directly, the programmer defines a procedure (the metaprogram) that, when run, computes a description of the machine. The metaprogramming tools have the property that they produce only real-time target machines. Of course, the metaprogram itself need not be real-time, since it is not intended to be coupled to the robot's physical environment.

In the remainder of this paper we present a brief description of the theoretical background of this work, an introduction to Rex with some simple examples of its application to problems suggested by the mobile robot domain, and a discussion of the synthesis problem.

2 Theoretical Background

A useful theory of intelligent embedded systems must be capable of describing how certain parts of the physical world encode information about other parts over time and how their behavior exploits that information. We model this situation abstractly by constructing the requisite concepts from a small set of primitives: space, time, possibility, and truth.

2.1 Basic Concepts

Let a universe $U = (L, T, W)$, where the set L (*locations*) is a topology suitable for modeling physical space, T (*times*) an ordered set of instants, and W (*possible worlds*) an abstract set of indices of possibility, i.e. possible histories or ways the world could be.

We define the set of *propositions* Φ to be $2^{W \times T}$. Intuitively, each element $\varphi \in \Phi$ is the set of world-time pairs in which that proposition holds. Φ has the structure of a Boolean algebra (of sets). The ordering $\sqsubseteq$ corresponds to entailment: $\varphi \sqsubseteq \varphi'$ means that φ is less general than (i.e., entails) φ'. The operations $\sqcap$, $\sqcup$, and $\neg$ correspond to intersection, union, and complementation of propositions. The *strongest postcondition* operator $\mathbf{S}^\epsilon : \Phi \rightarrow \Phi$ satisfies: $\mathbf{S}^\epsilon(\varphi)(w, t + \epsilon) \equiv \varphi(w, t)$. If the superscript ϵ is omitted, it is assumed to be equal to 1.

We identify *processes* with their spatial trajectories, i.e., the set of mappings $\pi : W \times T \rightarrow L$. $\pi(w, t)$ denotes the volume of space occupied by process π in world w at time t. The set of processes inherits the structure of L; it is closed under pointwise union, intersection, and complementation, and one process can be a subprocess of another. The null process is denoted by $[\ ]$, and $[\pi_1, \ldots, \pi_n]$ denotes a process tuple that is made up of subprocesses $\pi_1, \ldots, \pi_n$.

The *value domain* of a process π, written D_π, is a distinguished set of mutually exclusive and exhaustive properties of that process. For any process π, the function $val_\pi : W \times T \rightarrow D_\pi$ associates with each world and time the value (or state) of π in that world at that time. Two processes π_1 and π_2 are said to be *behaviorally equivalent* (written $\pi_1 == \pi_2$) if and only if they take on the same value at each world and time. Formally:

$$\pi_1 == \pi_2 \ \equiv \ \forall w, t.\, val_{\pi_1}(w, t) = val_{\pi_2}(w, t)$$

In designing computational systems, we are especially interested in *discrete* processes, i.e., processes that can be described in terms of discrete sets of locations, states, and instants of time. For example, registers in a digital computer are easily modeled as compound processes made up of flip-flop subprocesses with value domain $\{H, L\}$, H denoting the property of being in a high-voltage state and L a low-voltage state.

A *machine* is modeled as a pair of (possibly complex) discrete processes subject to behavioral constraints. The notation $m(X, Y)$ means that output process Y acts as a machine of type m with respect to input process X, i.e., X and Y satisfy the behavioral constraints imposed by m. When we wish to be concrete, we refer to these processes as *storage locations*, since they can be realized as physical components in digital hardware.

We shall make use of two varieties of primitive machine: pure functional machines (e.g., logic gates), symbolized by f^*, and delay elements, Δ_c. These machine types are characterized by the following formulas:

$$\begin{aligned} f^*(X,Y) &\equiv \forall w,t.\ val_Y(w,t) = f(val_X(w,t)) \\ \Delta_c(X,Y) &\equiv \forall w.\ (val_Y(w,0) = c) \\ &\qquad\qquad \wedge (\forall t \geq 0.\ val_Y(w,t+1) = val_X(w,t)). \end{aligned}$$

The f^* machine "instantaneously" computes the primitive function $f : D_X \rightarrow D_Y$; the output of the delay machine is the constant c, followed by its input, displaced in time by one unit. Complex machines are ultimately made up of storage locations constrained to act as machines of these primitive types and may be built up through the use of composition operators. One complete set of such operators consists of *serial*, *parallel*, and *feedback* compositions. These have well-understood mathematical properties and have been studied extensively in the context of the theory of automata and switching circuits [4].

2.2 The Information Content of Processes

In possible-world models of modal logics of knowledge with world-time indices, $w, t \models K(X, \varphi)$ is usually defined to be true if and only if $w', t' \models \varphi$ for all w', t' *epistemically accessible* to agent X from w, t. If the accessibility relation is an equivalence relation, the logic will satisfy the axioms of modal system S5 [6], including the axioms of deductive closure, positive introspection, and negative introspection. One approach to formalizing the information content of processes would be to use such a modal logic of knowledge, with agents identified with processes and the epistemic accessibility relation for a process π defined as follows:

$$w, t \approx_\pi w', t' \equiv val_\pi(w,t) = val_\pi(w',t')$$

Under this definition, $\approx_\pi$ is clearly an equivalence relation on $W \times T$, and the S5 axioms are satisfied [14].

In place of the $K(X, \varphi)$ notation, it will be convenient to make use of a *denotation function* that maps the values of a process to their propositional content. For a given process π and value $v \in D_\pi$, we define the denotation of v for π as the strongest proposition consistent with π's having value v. This proposition corresponds to the information that the process has about its environment when its value is v. We formally define the *denotation* function from values to propositions $\mu_\pi : D_\pi \to \Phi$ as

$$\mu_\pi(v) = \{(w,t) \mid val_\pi(w,t) = v\}.$$

Denotations and knowledge are directly related in the following way:

$$w,t \models K(\pi,\varphi) \equiv \mu_\pi(val_\pi(w,t)) \sqsubseteq \varphi$$

The ordering on Φ induces an ordering on denotation functions over the same value domain:

$$\mu_1 \sqsubseteq \mu_2 \equiv \forall v.\ \mu_1(v) \sqsubseteq \mu_2(v)$$

If $\mu_1 \sqsubseteq \mu_2$, then μ_1 is at least as informative as μ_2.

The need for formal semantics of knowledge representations is well recognized by AI researchers. Traditionally, however, denotation functions have been stipulated uniformly, in the sense that the same symbols are used in every module to mean the same things. Furthermore, the relation between the operation of the machine and the content of the representation is often ignored. In situated-automata theory, a more fine-grained approach to denotation is adopted. Meanings are associated to values in a location-dependent fashion, and the denotation function depends crucially on the behavior of the machine. The relationship between denotation and machine structure is the subject of the next section.

2.3 Machines as Inducers of Semantic Transformations

A machine can be seen as performing a transduction from the time series of values at its input location to values at its output location. Correspondingly, at the denotational level, each machine type has associated with it a higher-order function on denotation functions. We call this function the *semantic transformation function* of the machine; it takes the denotation function of the input onto the denotation function of the output. We will notate the semantic transformation function associated with machine type m by $\tau(m)$. Formally,

$$m(X,Y) \supset \mu_Y = \tau(m)(\mu_X).$$

For any machine, the semantic transformation function is entirely determined by the transformation functions of the primitive machines and the interconnection of the primitive machines.

For the pure functional machines f^*, the semantic transformation function is defined in the following way:

$$\tau(f^*)(\mu)(v) = \bigsqcup_{u \in f^{-1}(v)} \mu(u)$$

Essentially, the denotation function of a particular value of the output location of a functional machine is a disjunction over the denotations of all of the possible values of the input location which could have given rise to that value in the output location.

For Δ_c, the family of delay machines parameterized by c, the semantic transformation function is defined as follows:

$$\tau(\Delta_c)(\mu)(v) = \begin{cases} \varphi_0 \sqcup \mathbf{S}(\mu(v)) & \text{if } v = c \\ \mathbf{S}(\mu(v)) & \text{otherwise} \end{cases}$$

The proposition φ_0 is taken to be strongest proposition guaranteed to be true when the machine is started. Formally, $\varphi_0 = \{(w, 0) \mid w \in W\}$. The denotation of a value v at the output location is either the strongest postcondition of the denotation of v at the input location if $v \neq c$ or, if $v = c$, the disjunction of that proposition with φ_0.

The denotation function of a complex storage location $[X_1, \ldots, X_n]$ is the intersection of the denotation functions of its sublocations:

$$\mu_{[X_1,\ldots,X_n]}([u_1, \ldots, u_n]) = \prod_{1 \leq i \leq n} \mu_{X_i}(u_i).$$

It follows that information is spatially monotonic; if X is a subprocess of Y and X carries the information that φ, then so does Y. Of course the converse is not true in general, and much of the "inference" that goes on in an intelligent machine might be viewed as *information localization*, i.e., causing information carried by a large piece of storage to be carried by a smaller piece.

In addition, all semantic transformation functions induced by machines are monotonic. This can be seen by observing that no negations occur in the definition of any of the semantic transformation functions; intersection and union are both monotonic functions on the domain of denotation functions. Even an inverter (i.e. a primitive not^* where $not(0) = 1, not(1) = 0$) induces a monotonic semantic transformation function;

$$\mu_1 \sqsubseteq \mu_2 \supset \tau(not^*)(\mu_1) \sqsubseteq \tau(not^*)(\mu_2).$$

Also, it is not the case that $\tau(not^*)(\mu)(0) = \neg(\mu)(0)$; instead $\tau(not^*)(\mu)(0) = \mu(1)$, a different proposition entirely.

Given machines m_1 and m_2 and their corresponding semantic transformation functions, it is possible to calculate the semantic transformation functions of the compositions of these machines. Let $m_1 \circ m_2$ and $m_1 \parallel m_2$ respectively denote the *serial* and *parallel* compositions of m_1 and m_2, and let $\circlearrowright m$ denote the *feedback* operator applied to m.

The semantic transformation function of the serial composition of m_1 and m_2 is simply the function composition of the semantic transformation functions.

$$\tau(m_1 \circ m_2)(\mu)(v) = \tau(m_2)(\tau(m_1)(\mu))(v)$$

In the parallel case, the semantic transformation function of the composition of m_1 and m_2 satisfies the following equation:

$$\tau(m_1 \parallel m_2)(\mu)(v) = \tau(m_1)(\mu)(v) \sqcap \tau(m_2)(\mu)(v)$$

The feedback case involves a fixpoint. $\tau(\odot m)(\mu)$ satisfies the following:

$$\tau(\odot m)(\mu)(v) = \tau(m)(\mu')(v)$$

where $\mu'([u_1, u_2]) = \mu(u_1) \sqcap \tau(\odot m)(\mu)(u_2)$.

3 A Framework for Metaprogramming

Rex is a set of development tools for constructing complex machines hierarchically. Machines are built by defining a metaprogram that constructs an abstract machine description by creating storage designators and incrementally constraining them to behave in particular ways with respect to one another. This description, which stipulates how the value of each atomic storage location is to be computed over time, can then be transformed either into physical circuitry or into code that simulates the described machine. This process is depicted schematically in Figure 1.

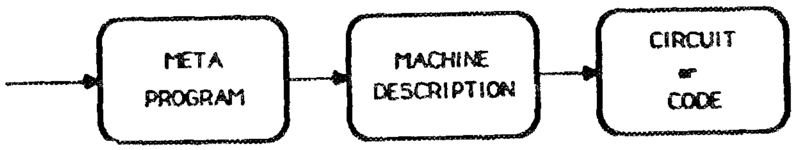

Figure 1: Stages in Machine Construction

3.1 A Description of Rex

In this section we present an informal description of the constructs that make up Rex, working from primitive machine constructors to the definition of arbitrarily complex machine constructors by the user.

3.1.1 Primitive Machine Constructors

There are two kinds of primitive machine constructor in Rex, corresponding to the primitive machine types discussed in Section 2. For the Δ_c machine, we have

(`init-next` *value expr*),

which denotes a storage location that is constrained to contain the value *value* initially and to contain at time $t+1$ the value at t of the location denoted by *expr*.

The family of primitive function machines,

(*primfn* $expr_1 \ldots expr_n$),

denote locations constrained to always contain the result of applying *primfn* to the the values of the locations denoted by $expr_1 \ldots expr_n$. *primfn* may be any one of a set of primitive machine constructors available to the programmer. By convention, an identifier *name*`m` names a machine constructor intuitively related to the function *name*, for example `ifm`, `timesm`, `squarem`, `equalm`, and `cosm`.

3.1.2 Storage Expressions and Wffs

We refer to invocations of constructors that denote storage locations, such as the ones mentioned above, as storage expressions. These may have behavioral constraints associated with them arising from their construction. A storage expression with no behavioral constraints may be created from an identifier with the form

(`stg` *ident*).

The form

(`if` *condition* $expr_1$ $expr_2$)

allows the structure of the machine to depend on conditions that are evaluated at construction time. If *condition* is true at the time this machine constructor is invoked, this expression denotes $expr_1$, else $expr_2$. In order to impose complex constraints on the behavior of a particular storage location, we use the form

(`the` *var* $wff_1 \ldots wff_n$),

which binds *var* to a new storage location, constrains that location to satisfy $wff_1 \ldots wff_n$, and returns it.

A *wff*, or well-formed formula, serves to constrain the behavior of the storage locations it mentions, but does not denote a particular storage location. There are four wff forms in Rex. The first form,

(== $expr_1$ $expr_2$)

constrains the storage locations denoted by $expr_1$ and $expr_2$ to be *behaviorally equivalent*, which means that at every point in time, each is to contain the same value as the other. It is a programming error to attempt to constrain two storage locations to be behaviorally equivalent if they are already constrained to behave in a way that precludes this possibility. The actual implementation of Rex imposes the slightly stronger requirement that at least one of the expressions in a == form must denote an unconstrained storage location. The form

(if *condition* wff_1 wff_2) ,

like the if form in the previous paragraph, depends on the value of *condition* at the time of invocation. If it is true, this form imposes the constraints of wff_1, otherwise, wff_2. The form

(and $wff_1 \ldots wff_n$)

imposes the conjunction of constraints of $wff_1 \ldots wff_n$. The form

(some ($var_1 \ldots var_k$) $wff_1 \ldots wff_n$)

is similar to the (the ...) form, generating k new storage locations and imposing multiple constraints upon them but returning no storage designator.

3.1.3 Defining New Machine Constructors and Constrainers

In Rex, both machine constructors and machine constrainers may be hierarchically defined. These correspond to storage expressions and wffs, respectively. The form

(defm *name* {$param_1 \ldots param_n$} ($arg_1 \ldots arg_m$) *expr*)

binds the identifier *name* to its definition as a machine constructor. The braces hold a list of parameters that are used by the machine constructor at construction time. The arguments denote the input locations of the machine under construction; *expr* denotes the (possibly compound) output storage location and constrains its behavior. The form

(defr *name* {$param_1 \ldots param_n$} ($arg_1 \ldots arg_m$) $wff_1 \ldots wff_n$)

is much like the defm form, binding *name* to its definition as a machine constrainer. The constraints it will impose are the conjoined constraints of $wff_1 \ldots wff_n$.

Once a machine constructor or constrainer *name* has been defined through the use of defm or defr, it may be invoked as a storage expressionn or wff as follows:

$$(name\ actual\text{-}param_1 \ldots\ actual\text{-}param_n\ actual\text{-}arg_1 \ldots\ actual\text{-}arg_m).$$

All of the storage locations that we have discussed so far are atomic; however, just as the logic admits of compound processes (see Section 2), storage expressions in Rex can denote compound objects. The expression [x . y] denotes the storage location which is the pair of locations denoted by x and y. As in Lisp, tuples (or lists) are built up from pairs, with [x_1 . [x_2 . ... [x_n . [ ]] ...]] abbreviated as [x_1 x_2 ... x_n]. Through the use of the == operator, storage designators can be *unified*, allowing compact metaprograms to recursively instantiate storage and constrain its behavior. (See the example in Section 3.5.)

Syntactically, we allow structured arguments in the argument list of a defm form. These are handled by unifying them with the actual arguments when the function is invoked. Thus,

```
(defm f {} ([x y z])
  (g x y z))
```

is equivalent to

```
(defm f {} (u)
  (the v
    (some (x y z)
      (== u [x y z])
      (== v (g x y z)))))
```

We also allow the first argument of a the form to be structured, in order to simplify writing expressions which denote complex storage. Thus, we can write

```
(the [x y z]
  (== x (f y z))
  (== y (f x z))
  (== z (f x y)))
```

rather than

```
(the w
  (== w [x y z])
  (== x (f y z))
  (== y (f x z))
  (== z (f x y)))
```

The notation %c is used as an abbreviation for the constant c machine constructor and is equivalent to

```
(the x (== x (init-next c x))).
```

We note that constants may have complex structure if the structure of the initial value is the same as that of the storage containing the constant. Thus, the following expression denotes a valid constant containing the triple $[1, 2, 3]$:

```
(the [x y z] (== [x y z] (init-next '(1 2 3) [x y z])))
```

3.2 Running Sum of Squares

As a simple introductory example, we describe the process of constructing a machine that continually computes the running sum of the squares of its inputs over time. A Rex function that constructs a description of this machine can be defined as follows:

```
(defm running-sum-of-squares () (input)
  (the sum
    (== sum (init-next 0 (plusm sum (squarem input))))))
```

Rex takes the preceding text as input, and generates the following abstract machine description (a schematic diagram of this machine is given in Figure 2):

```
((PLUS T1 T2 PLUS1)
 (SQUARE INPUT T1)
 (DELAY 0 PLUS1 T2))
```

In the abstract machine description, the atoms INPUT, PLUS1, T1, and T2 designate storage locations, which, for example, in a digital circuit, would be wires carrying signals. The machine description can be interpreted in two ways. From the structural point of view, each line corresponds to a primitive component of the machine and the description as a whole encodes the connectivity of the components. From the behavioral point of view, each line of the description imposes constraints on the behavior of the storage locations it mentions.

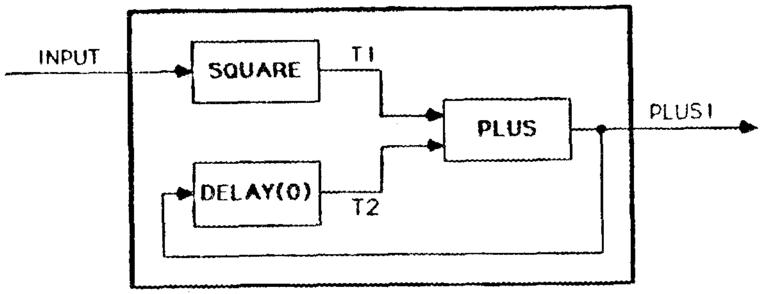

Figure 2: A Simple Machine for Computing Running Sum of Squares

Structurally, (PLUS T1 T2 PLUS1) means that there is an "adder" permanently connecting input locations T1 and T2, with output location PLUS1. (SQUARE INPUT T1) means that location T1 will be the output location of a "squaring' component with input location INPUT. The behavioral interpretations of PLUS and SQUARE are self-evident. Structurally, (DELAY 0 PLUS1 T2) means that a delay element connects location PLUS1 with location T2. Behaviorally, T2 has the initial value 0 and at time $t+1$ has the value of PLUS1 at time t. The reader can easily verify that, at any point in time, the location PLUS1 contains the sum of the squares of all the previous values of INPUT.

As a (trivial) illustration of the use of construction-time parameters, we redefine running-sum as follows:

```
(defm running-sum-of-squares {init} (input)
  (the sum
    (== sum (init-next init (plusm sum (squarem input))))))
```

This parameterized version of the definition allows Rex to construct a family of machines, all of which add a quantity to the running sum but differ in the quantity they add.

3.3 Machine Compositions in Rex

Various forms of machine composition (eg. serial composition, parallel composition, and feedback) are expressed naturally in Rex. (See Figure 3 for the schematics.) Serial composition corresponds to simple function composition,

```
(defm f {} (x) (g (h x)));
```

parallel composition is achieved through pairing,

```
(defm f {} (x) [(g x) . (h x)]);
```

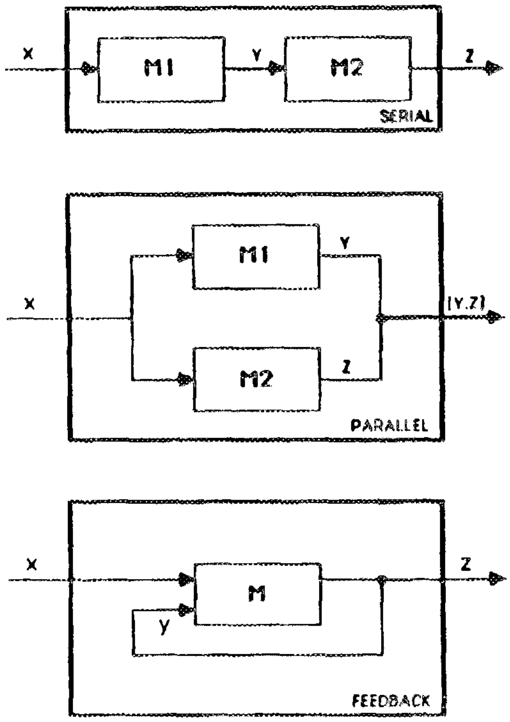

Figure 3: Serial, Parallel, and Feedback Composition

and feedback comes about through cyclically dependent variables.

```
(defm f {} (x)
  (the y
    (== y (g x y)))).
```

By using higher-order definitions in which machine constructors are parameters of other machine constructors, these compositions can be defined generically, though they are not often used in this form. The definitions are as follows[1]:

```
(defm serial {m1 m2} (x) (m2 (m1 x)))

(defm parallel {m1 m2} (x)  [(m1 x) . (m2 x)]))

(defm feedback {m} (x)
  (the y
    (== y (m x y))))
```

[1]If the implementation is in a version of Lisp that requires `funcall`, these definitions must be modified slightly to include the call explicitly.

3.4 Position and Orientation

This section contains an example illustrating how Rex can be conveniently used in the mobile-robot domain to construct machines that track certain properties of the environment. The machine portrayed schematically in Figure 4 is intended to be a submodule of a mobile-robot program. If the machine's input location tracks the motor-command output of the entire robot, its output will track the robot's position and orientation with respect to its initial position and orientation. The new position and orientation are functions of the old position and orientation and of the current action as well. The entire machine is a serial-parallel composition of two submachines, one for orienation and another for position, with the output of the orientation machine constituting one of the inputs to the position machine. Note that reversing the sequence of the == expressions would have no effect on the generated machine since constraints can accumulate in any order.

In Figure 5 we present the definitions of the various modules of the position and orientation machine constructors. At the submodule level, the structure of the action is broken down into a command and an argument. The command may be either `turn`, `forward`, or `noop`. If the command is `turn`, the argument is the number of degrees the robot is turning; if it is `forward`, the argument is the distance the robot is moving forward. The argument carries no information if the command is `noop`.

In the Rex definitions, the locations `x`, `y`, and `orient` always contain the current position and orientation, while the local variables `local-x`, `local-y`, and `local-orient` are used to store values for the next computational step.

3.5 Prioritized-choice Machine

As another example from the mobile robot domain, consider a mobile robot that is intended to carry out many tasks in parallel but with differing priorities. As a concrete illustration, let us imagine that the robot is supposed to avoid collisions, take the second possible left turn, stay parallel with the wall on its right, and keep moving, in that order of priority. The example in Figure 6 shows how Rex is used for prioritizing such activities. For simplicity, in this example the priorities are frozen at construction time; dynamic prioritization can also accomplished within the Rex framework by encoding priorities in the state of the machine.

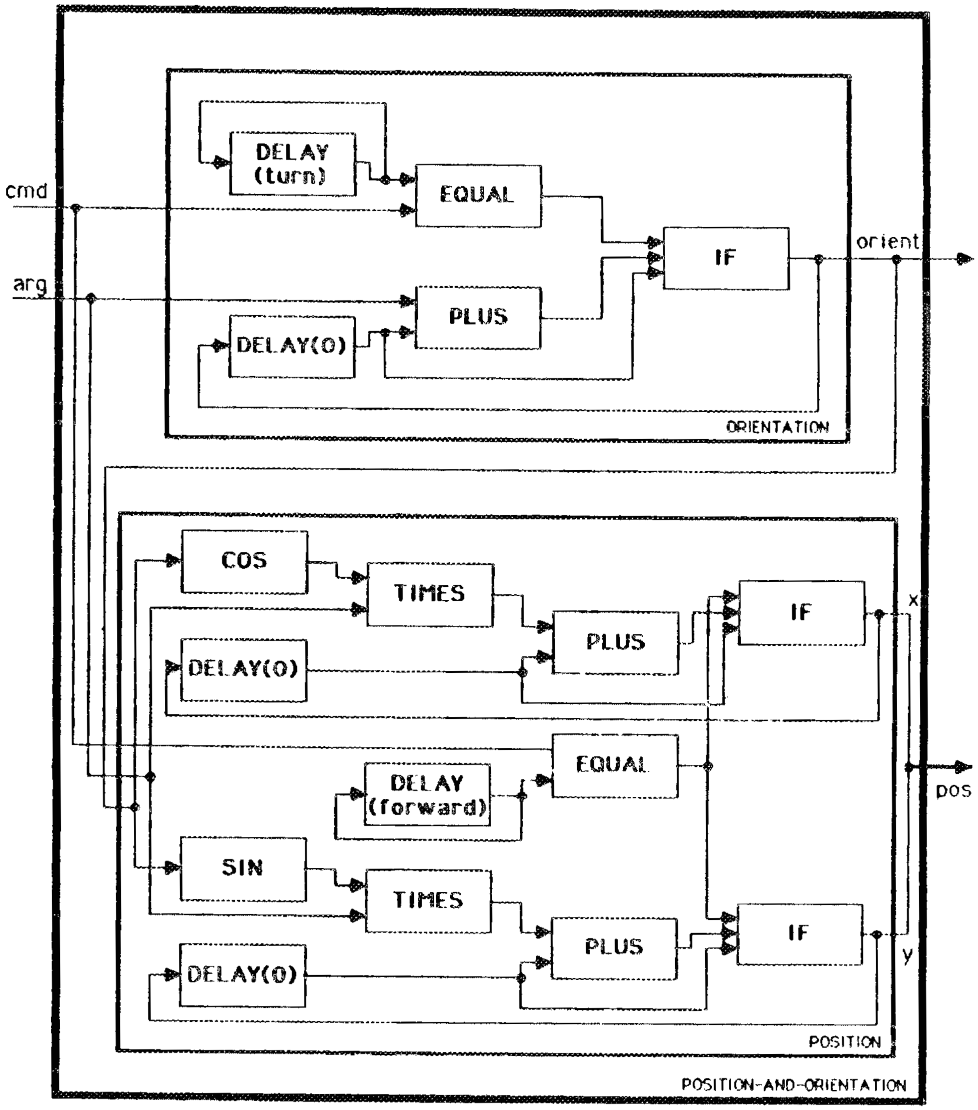

Figure 4: Schematic Diagram of Position and Orientation Machine

```
(defm position-and-orientation {} (action)
  (the [pos orient]
    (== orient (orientation action))
    (== pos (position action orient))))

(defm orientation {} ([cmd arg])
  (the orient
    (some (local-orient)
      (== local-orient (init-next 0 orient))
      (== orient (ifm (equalm cmd %turn)
                      (plusm local-orient arg)
                      local-orient)))))

(defm position {} ([cmd arg] orient)
  (the [x y]
    (some (local-x local-y)
      (== [local-x local-y] (init-next '(0 0) [x y]))
      (== x (ifm (equalm cmd %forward)
                 (plusm local-x (timesm arg (cosm orient)))
                 local-x))
      (== y (ifm (equalm cmd %forward)
                 (plusm local-y (timesm arg (sinm orient)))
                 local-y)))))
```

Figure 5: Rex Definitions for Position and Orientation Machine Constructors

```
(defm robot {} (data)
  (priority-choose 4 [(avoid-collision data)
      (second-left-turn data)
      (parallel-to-right-wall data)
      (keep-moving)]))

(defm priority-choose {n} (choice-list)
  (the choice
    (if (= n 1)
        (== choice-list [choice])
        (some (head tail)
          (== choice-list [head . tail])
          (== choice (ifm (equalm head %noop)
                          (priority-choose (- n 1) tail)
                          head))))
```

Figure 6: Rex Definition of Prioritized-choice Machine Constructors

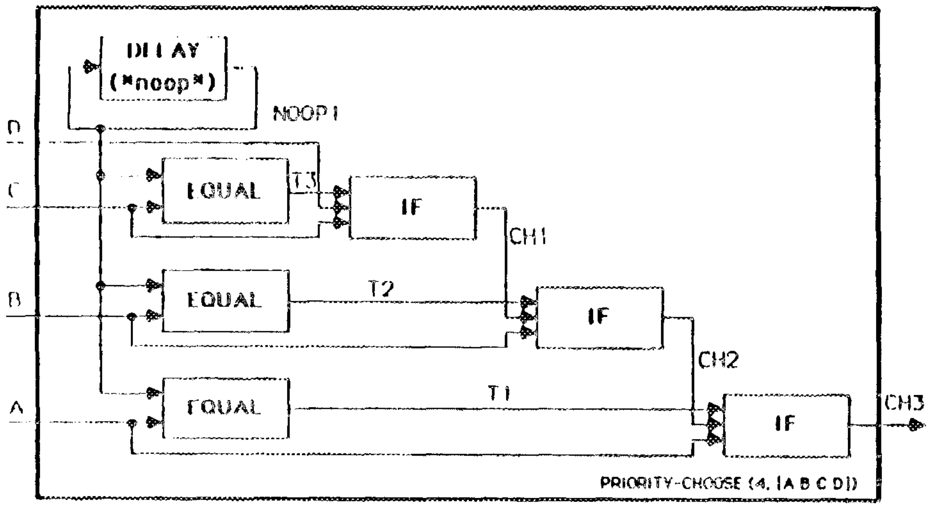

Figure 7: Schematic diagram of prioritized choice machine

```
        (EQUAL NOOP1 A T1)
        (EQUAL NOOP1 B T2)
        (EQUAL NOOP1 C T3)
        (IF T3 D C CH1)
        (IF T2 CH1 B CH2)
        (IF T1 CH2 A CH3)
        (DELAY *NOOP* NOOP1 NOOP1)
; *NOOP* is the (constant) value of NOOP1
```

Figure 8: Machine Description of Prioritized-choice Machine

The robot machine constructor builds a machine that takes data as input and generates actions. We shall assume we already have four machines that transduce values on the data line into either actions or the value noop. The output value of each machine indicates what the robot must do to satisfy that machine's goal. The metaprogram connects these machines together in such a way as to cause the resultant overall machine to deliver as output the value of the highest-priority submachine whose output is different from noop. If there is no such value, the last action is output, whether or not its value is noop. We emphasize that the recursion occurs at construction time and results in a spatial array of components rather than a temporal succession of computational steps.

Figure 8 contains the linearized abstract machine description computed by using the invocation (priority-choose 4 [a b c d]). The storage location returned is CH3.

4 Reasoning about Epistemic Properties

4.1 Analysis

The theoretical concepts presented in Section 2 can be used to analyze the semantic properties of machines. The theory determines how the denotation of the outputs depend on the denotations of the inputs and the structure of the machine. In practice, when these functions become complex, the designer may find it convenient to specify the denotation function μ indirectly by positing a convenient auxiliary domain A and expressing μ as the composition of two functions d and e, where $d : D_X \to A$, $e : A \to \Phi$, and $\mu(v) = e(d(v))$.

The following is an example illustrating the use of auxiliary denotation functions for structured data domains. Let $Y = [P, N]$ be a compound process, where $D_P = \{\mathbf{man}, \mathbf{boy}, \mathbf{woman}, \mathbf{girl}\}$ and $D_N = \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots\}$. (The bold typeface is to emphasize that the symbols are to be regarded as simple data values.) Let A_1 be some set of properties of individuals, A_2 the set of natural numbers, and let $d_1(\mathbf{man}) = man$, etc., and $d_2(\mathbf{n}) = n$. Then, we can define

$$e([p, n]) = \{(w, t) \mid \exists a.\, p(a)(w, t) \wedge age(a, w, t) = n\}$$

and set

$$\mu_Y([u, v]) = e([d_1(u), d_2(v)]).$$

This definition implies, for example, that

$$\mu_Y([\mathbf{girl}, \mathbf{7}])(w, t) \equiv \exists a.\, girl(a)(w, t) \wedge age(a, w, t) = 7.$$

Notice that if men are constrained to be over twenty-one years of age, then if m is a machine whose semantic transform is μ_Y, it is a *theorem* that Y never takes on the value $[\mathbf{man}, \mathbf{7}]$!

4.2 Example of Analysis

We use the tools introduced in previous sections to analyze the semantic properties of the `orientation` machine defined in Figure 4.

Since the constraints imposed by machines are inherently relational, it is much easier to prove properties of machines constructed by Rex if the defining forms are translated into a relational version of Rex. The translation is straightforward, and may be automated. In relational Rex, a colon is prefixed onto each function name, making it into a relational form. For example, `(== x (plusm y z))` is expressed in the relational form as `(:plusm y z x)`. The other salient difference is that in a `:defm` form, the output locations are listed as a fourth argument.

The relational version of the orientation module of the position and orientation machine (see Figure 4) is:

```
(:defm orientation {} ([cmd arg]) orient
   (some (t1 t2)
     (:init-next 0 orient local)
     (:plusm arg t1)
     (:equalm cmd %turn t2)
     (:ifm t2 t1 orient)))
```

This form of Rex is significantly more tedious for a programmer to use, due to the necessity of naming all intermediate storage locations.

The relational version of the Rex definition may then be translated into the logic, expressed as a conjunction of primitive machine constraints.

$$\begin{aligned}&\mathtt{orientation([cmd,arg],\ orient)} \equiv \\ &\quad \exists \mathtt{local, t1, t2}. \\ &\qquad \Delta_0(\mathtt{orient, local}) \wedge \\ &\qquad \mathit{plus}([\mathtt{local, arg}], \mathtt{t1}) \wedge \\ &\qquad \mathit{equal}([\mathtt{cmd}, \%\mathtt{turn}], \mathtt{t2}) \wedge \\ &\qquad \mathit{if}([\mathtt{t2, t1, local}], \mathtt{orient})\end{aligned}$$

Now, we give the denotation functions of the inputs, and derive the denotation function of the output. The value domains of the input components are:

$$D_{\mathtt{cmd}} = \{\mathbf{forward}, \mathbf{turn}, \mathbf{noop}\}$$

$$D_{\mathtt{arg}} = \{\ldots, -\mathbf{2}, -\mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots\}$$

The denotation function of the storage location cmd is as follows:

$$\mu_{\mathtt{cmd}}(\mathbf{forward}) \equiv \mathit{moving}$$

$$\mu_{\mathtt{cmd}}(\mathbf{turn}) \equiv \mathit{turning}$$

$$\mu_{\mathtt{cmd}}(\mathbf{noop}) \equiv \mathit{still},$$

where $moving(w,t) \supset \neg turning(w,t)$, etc. The denotation function of **arg** is most conveniently described as the composition of two functions, as discussed above. $\mu_{\mathtt{arg}}(\mathbf{n}) = e(d(\mathbf{n}))$ where $d(\mathbf{n}) = n$ and

$$e(n)(w,t) \equiv \begin{array}{l}(moving(w,t) \wedge dist(w,t) = n \wedge angle(w,t) = 0) \vee \\ (turning(w,t) \wedge dist(w,t) = 0 \wedge angle(w,t) = n) \vee \\ (still(w,t) \wedge dist(w,t) = 0 \wedge angle(w,t) = 0).\end{array}$$

The denotation function of orient can be derived from the formal description of the orientation machine and the input denotations:

$$\mu_{\mathtt{orient}}(v) = \{(w,t) \mid d(v) = \sum_{t'=0}^{t} angle(w,t')\}$$

In other words, the storage location `orient` always encodes the sum of the angles turned through by the robot since it was started.

4.3 Observations on Synthesis

The metaprogramming approach described in the Section 3 lends itself to the synthesis of machines with formally specifiable knowledge properties. To this point we have been considering how, given a denotation function of X and a machine $m(X,Y)$, one can compute the denotation function of Y. In practice, however, we are often interested in the inverse problem, namely, given the denotation function μ_X of the input and an *intended* denotation μ_Y for the output, construct a machine m that guarantees that μ_Y is indeed the objective denotation of the output. Formally, find an m such that

$$m(X,Y) \supset \mu_Y = \tau(m)(\mu_X).$$

It is difficult to guarantee exact equality in the general case; a more practical goal is to synthesize a machine that induces a denotation function satisfying specified properties. For example, we may wish to bound the induced denotation function above and below under the ordering $\sqsubseteq$ introduced in Section 2. That is, given an input denotation function μ and a pair of bounding denotation functions μ^- and μ^+ we might be interested in constructing a machine m such that

$$\mu^- \sqsubseteq \tau(m)(\mu) \sqsubseteq \mu^+.$$

Lower bounds guarantee "ignorance" while upper bounds guarantee "knowledge." (Guaranteeing ignorance can be a positive goal of the designer, e.g. in assuring the privacy of information in data bases.)

In cases where the notion of knowledge is too strong, a weaker notion similar to belief can be defined in terms of positive and negative knowledge conditions. This will allow us to build machines that "jump to conclusions" based on lack of knowledge and automatically retract them as new knowledge is gained. For example, working in a modal language, we can introduce axioms like the following for each specific φ of interest:

$$B(X,\varphi) \equiv K(X,\varphi) \vee (\neg K(X,\varphi) \wedge \neg K(X,\neg\varphi) \wedge B(X,\varphi'))$$

where φ' is a condition which provides sufficient evidence for X to believe φ. Eventually, the conditions ground out in positive and negative K formulas. $B(X,\varphi)$ is clearly nonmonotonic; increased information can falsify the B condition.

4.4 Compiled Knowledge

It is possible to employ static structures ordinarily used at run time to control the construction of efficient specialized machines. In Rex this amounts to defining a machine constructor

```
(defm robot {knowledge-base} (input) ...)
```

instead of

```
(defm robot {} (knowledge-base input) ...),
```

or similarly,

```
(defm parser {grammar} (input) ...)
```

instead of

```
(defm parser {} (grammar input) ...)
```

It should be noted that truly static processes, such as an unchanging assertional database or fixed grammar rules, carry no information beyond φ_0 and hence may be encoded directly as constraints among those processes that do vary over time.

Specific strategies exist for constructing machines that realize inference rules. Let us consider a pair of processes X and Y with value domains D_X and D_Y. Let the denotation function of X be μ_X and the ***intended*** denotation function of Y be μ and assume that D_Y forms a lattice under the ordering $\leq$ where

$$v_1 \leq v_2 \equiv \mu_Y(v_1) \sqsubseteq \mu_Y(v_2).$$

For any element $u \in D_X$, there is a unique greatest lower bound v of the set $\{v' \mid \mu_X(u) \sqsubseteq \mu(v')\}$ that is also a member of the set. We take $f : D_X \rightarrow D_Y$ to be the function that picks out this v; the function machine $f^*(X,Y)$ guarantees that the objective denotation function μ_Y entails the intended denotation function μ.

$$\mu_Y = \tau(f^*)(\mu_X) \sqsubseteq \mu$$

A similar construction can be performed for delay machines, and multiple machines can be combined uniformly by again taking greatest lower bounds of their results, provided their value domains have the structure of a lattice.

This idea can be exploited using the machinery presented in Section 3. In Rex, machine constructors may be parameterized by other machine constructors. This facility can be used to define generic modules that take as parameters machine constructors which embody particular inference rules. The generic module constructs a composite machine that at each point in time combines the results produced by the separate inference machines. The denotation of any value generated by the composite machine is guaranteed to be the strongest representable proposition that follows from the results of the individual inference modules. The following is an example of a generic machine of this sort. The parameters are each machine constructors: the semantic transform associated with each **infrule** parameter corresponds to a rule of inference; the **glb** parameter constructs a machine which takes greatest lower bounds in the lattice which is the target domain of the inference rules, as described above.

```
(defm combine-infrules {infrule1 infrule2 infrule3 glb} (data)
   (glb (infrule1 data)
        (glb (infrule2 data)
             (infrule3 data))))
```

5 Related Work

Our approach is similar in spirit to work by Johnson [7] on the synthesis of digital circuits from recursion equations. Johnson's work is based on the transformation of recursive behavioral specifications of a circuit into realizations. Similar methods have also been used by Hillis and Chapman for circuit design [1], and by Goad for model-based vision [2]. Rex also bears some resemblance to dataflow languages, e.g. Lucid [15], although our semantics are location-oriented rather than stream-oriented as in Lucid and other data-flow languages.

6 Implementation Status

The Rex system has been implemented in Zetalisp and Common Lisp and is currently running on the Symbolics 3600, DEC 2060, and Sun Workstation. Rex is implemented as an extension to Lisp making use of Lisp's macro facility for special syntactic forms. Rex definitions result in the creation of Lisp functions that construct machine descriptions by collecting and propagating constraints on storage locations of the target machine. Equational constraints are resolved by using a variant of the unification algorithm. An abstract machine description computed by Rex may be realized in digital hardware, since it is virtually a circuit diagram and seems well suited for implementation on fine-grained parallel architectures such as the Connection Machine. However, it is also suitable for realization as code in conventional languages for sequential hardware. Our current implementation, for instance, supports code generation in both Lisp and C. The congruence closure algorithm is employed to eliminate common subcomputations; a topological sort is performed to order variable assignments (storage location updates) according to data dependency in the abstract machine.

The Rex environment is presently being used to implement complex robot control programs for SRI's mobile robot.

7 Acknowledgments

This work was supported in part by a gift from the Systems Development Foundation, in part by FMC Corporation under contract 147466 (SRI Project 7390), and in part by General Motors Research Laboratories under contract 50-13 (SRI Project 8662).

We have profited greatly from discussions with David Chapman.

8 REFERENCES

[1] Chapman, David (personal communication).

[2] Goad, Chris. "Special Purpose Automatic Programming for 3D Model-based Vision." Proceedings of Image Understanding Workshop, Arlington, VA, 23 June, 1983, pp. 94-104.

[3] Halpern, Joseph and Y.O. Moses. "Knowledge and common knowledge in a distributed environment." Proceedings of the 3rd ACM Conference on Principles of Distributed Computing, 1984, pp. 50-61; a revised version appears as IBM RJ 4421, 1984.

[4] Hartmanis, J. and R.E. Stearns. *Algebraic Structure Theory of Sequential Machines*. Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1966.

[5] Hintikka, J. *Knowledge and Belief.* Cornell University Press, Ithaca, 1962.

[6] Hughes, G. E. and M. J. Cresswell. *An Introduction to Modal Logic*. Methuen and Co. Ltd., London, 1968.

[7] Johnson, Steven D. *Synthesis of Digital Designs from Recursion Equations*. MIT Press, Cambridge, Massachusetts, 1984.

[8] Konolige, Kurt. *A Deduction Model of Belief and its Logics*. Technical Note No. 326, Artificial Intelligence Center, SRI International, Menlo Park, CA, August, 1984.

[9] Kripke, Saul. "Semantical Analysis of Modal Logic." *Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik* **9**, 1963, pp. 67-96.

[10] Levesque, Hector J. "A Logic of Implicit and Explicit Belief." Proceedings of the National Conference on Artificial Intelligence, 1984, pp. 198-202.

[11] McCarthy, John. "Programs with Common Sense." In *Semantic Information Processing*, Marvin Minsky (ed.), MIT Press, Cambridge, Massachusetts, 1968.

[12] Moore, Robert C. "A Formal Theory of Knowledge and Action." In *Formal Theories of the Commonsense World*, Jerry R. Hobbs and Robert C. Moore (eds.), Ablex Publishing Company, Norwood, New Jersey, 1985.

[13] Nilsson, Nils J., "Shakey the Robot," Technical Note No. 323, Artificial Intelligence Center, SRI International, Menlo Park, CA, April 1984.

[14] Rosenschein, Stanley J. "Formal Theories of Knowledge in AI and Robotics." Proceedings of Workshop on Intelligent Robots: Achievements and Issues," SRI International, Menlo Park, CA, 13-14 November, 1984, pp. 237-252.

[15] Wadge, William W. and Edward A. Ashcroft. *Lucid, the Dataflow Programming Language*. Academic Press, London, 1985.

Paper 3

AN INTELLIGENT MACHINE OPERATING SYSTEM FOR HYPERCUBE ENSEMBLE ARCHITECTURES*

J. Barhen

Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
P.O. Box X, Oak Ridge, TN 37831

ABSTRACT

The research we conduct for the Department of Energy [Office of Basic Energy Sciences], the United States Air Force [Wright Aeronautical Laboratories] and the United States Army [Human Engineering Laboratory] involves the development of dynamic resource allocation (scheduling and load-balancing) algorithms in a virtual time environment. These algorithms will be embedded into a virtual time intelligent machine operating system. Our emphasis is on applications characterized by structures irregular in time and space, with irregularities unpredictable in advance, and which might be as often communication-bound as compute-bound. Since our ultimate objective is to exploit advanced computer architectures for machine intelligence problems, the generic IMOS/VT methodology is targeted at a wide spectrum of concurrent computation requirements, extending from "coarse-grain" architectures to "fine-grain" connection-machine-type systems. Our current implementation framework focuses on hypercube ensembles.

I. INTRODUCTION

For the successful real-time operation of a wide range of autonomous or semi-autonomous intelligence-targeted robotic systems, it is essential that the computers on board be able to "think" fast enough. The current consensus is that while the microprocessors at the heart of any computer

*This work is currently funded by the U.S. Department of Energy (Office of Basic Energy Sciences), the U.S. Army (Human Engineering Laboratory) and the U.S. Air Force (Wright Aeronautical Laboratories).

will probably not become much faster, there is a continuing trend for them to become smaller and cheaper. Thus the key to more powerful computers (i.e., faster thinking) is to have many processors cooperating in the solution of a given problem. Such systems are defined as "concurrent" rather than parallel, to avoid the "lockstep" connotation associated with the latter.

The development of concurrent computers, particularly in the context of intelligent machines, raises several challenging issues. How powerful should each processor be? How should the processors communicate with each other? How should the workload be divided among the processors? How does one make sure that processors are not sitting idle waiting for input from other processors? The Center for Engineering Systems Advanced Research (CESAR) at the Oak Ridge National Laboratory (ORNL) has recently initiated a program which starts from some of the most advanced and promising developments in concurrent computation. It addresses research required to develop an efficient systems' environment including dynamic resource allocation (i.e., load-balancing/scheduling) algorithms within a virtual-time operating system suitable for a wide range of real-time applications.

The computer design being investigated at ORNL/CESAR is based on a "hypercube" architecture. The system, built by NCUBE Corporation, was designed from the ground up to be optimally implemented in state-of-the-art VLSI. It provides unmatched raw performance since up to 1024 processors, each of about the power of one and a half VAX 11/780, can be connected to their nearest boolean hypercube neighbors and communicate only through message passing. Since VLSI technology is used, the total volume of such a system is much less than one cubic meter. Recent research at the California Institute of Technology (Caltech) has shown a similar design to be one of the most powerful and versatile.

Our ultimate intent is to develop a "Virtual Time" Intelligent Machine Operating System (IMOS/VT), to provide a generalized framework for implementing machine intelligence. This type of operating system is expected to be especially suitable for hard-real-time environments, as encountered in autonomous machines or SDI applications, since processors will be able to think ahead in "virtual time", issue a set of tentative commands, and modify them only if new information warrants it. It is this thinking ahead which, for problems involving thousands of processors and of processes with a time-varying interconnection structure, evens out the workload not only in time but also between processors.

Currently we are involved with the development of a "generic" version of IMOS/VT and supporting algorithms. In particular, since the emphasis is on exploiting advanced computer architectures for machine intelligence applications, the virtual time methodology needs to be targeted at a wide spectrum of concurrent computation requirements, extending from "coarse-grain" architectures (e.g., the ORNL/NCUBE hypercube, the BBN butterfly multiprocessor) to "fine-grain" connection-machine-type systems. Development of a few selected IMOS/VT modules has just been initiated, using both the NCUBE machine and an ORNL-enhanced version of the Caltech hypercube simulator. This effort represents the first basic steps towards the goal of successful operation of complex distributed systems in hard real-time environments.

In Section II we outline some of the critical issues related to the control of intelligent machines using distributed concurrent processors. Design considerations in the development of a virtual-time operating system and its supporting algorithms are addressed in Section III and IV. We conclude by indicating key mile stones for a "full-fledged" implementation of the Virtual-Time Methodology.

II. CONTROL OF INTELLIGENT MACHINES USING CONCURRENT PROCESSORS: CRITICAL ISSUES

Advanced autonomous robots, such as the HERMIES-II prototype currently being developed and tested at CESAR[1] or the Hexapod walking machine constructed by Ohio State University,[2] and other intelligence-targeted machines of the future[3] are generally composed of a variety of asynchronously controlled components. For a robot, these components may include manipulator arms, electro-optical sensors, sonars, navigation controllers, etc... In order to take advantage of the distributed nature of the associated robotic processes, it was envisioned[4] that a Robot Operating System (ROS) should be developed, to provide a generalized framework for implementing machine intelligence, through real-time control of a distributed multimicroprocessor system. In the following, we first review some recent advances in message-passing architectures; we then address some critical issues of specific interest to this workshop.

a. VLSI-Based Message-Passing Ensembles

The rationale for our approach lies in recent advances in VLSI technology[5,6] which dramatically reduce the cost of computation. The basic trend is to use the state-of-the-art VLSI to integrate an entire processing system on a single chip, including communications links, memory interface and 32-bit processor, resulting in smaller and cheaper processors comparable in performance to their larger and more expensive predecessors. This trend, which we see as continuing over the next decade, is the major technological drive behind concurrent computation, i.e., the use of an ensemble of small computers that work concurrently on parts of a complex problem, and coordinate their computations entirely by sending messages to each other. Such considerations have recently led to the successful development of several families of such "ensemble machines." Work at Caltech, for example ranges in scope from the "cosmic cube" (initially 64 nodes connected in a Boolean 6-cube, using 8086/8087 16-bit processors and currently upgraded to 68020's) to the "mosaic experiment" (which involves single-chip nodes). In a similar vein, the much finer grain "connection machine" being developed by Thinking Machines Inc. is reported[7] to have implemented processor-to-processor communication through a fast message routing system that forms a hypercube. The intrinsic characteristics of hypercube ensembles, which are briefly summarized below, when put in perspective of the ROS requirements, provided us with strong incentives to configure the "brain" of future HERMIES robots as homogeneous hypercubes of appropriate dimensionality.

By "hypercube ensemble machine" we refer to a Multiple Instruction Multiple Data (MIMD) multiprocessor design in which $N=2^d$ identical (i.e., homogeneous) nodes are connected in a binary d-dimensional cube topology using fully asynchronous bidirectional channels. For illustrative purposes, a few hypercubes of low order are shown in Fig. 1, where circles

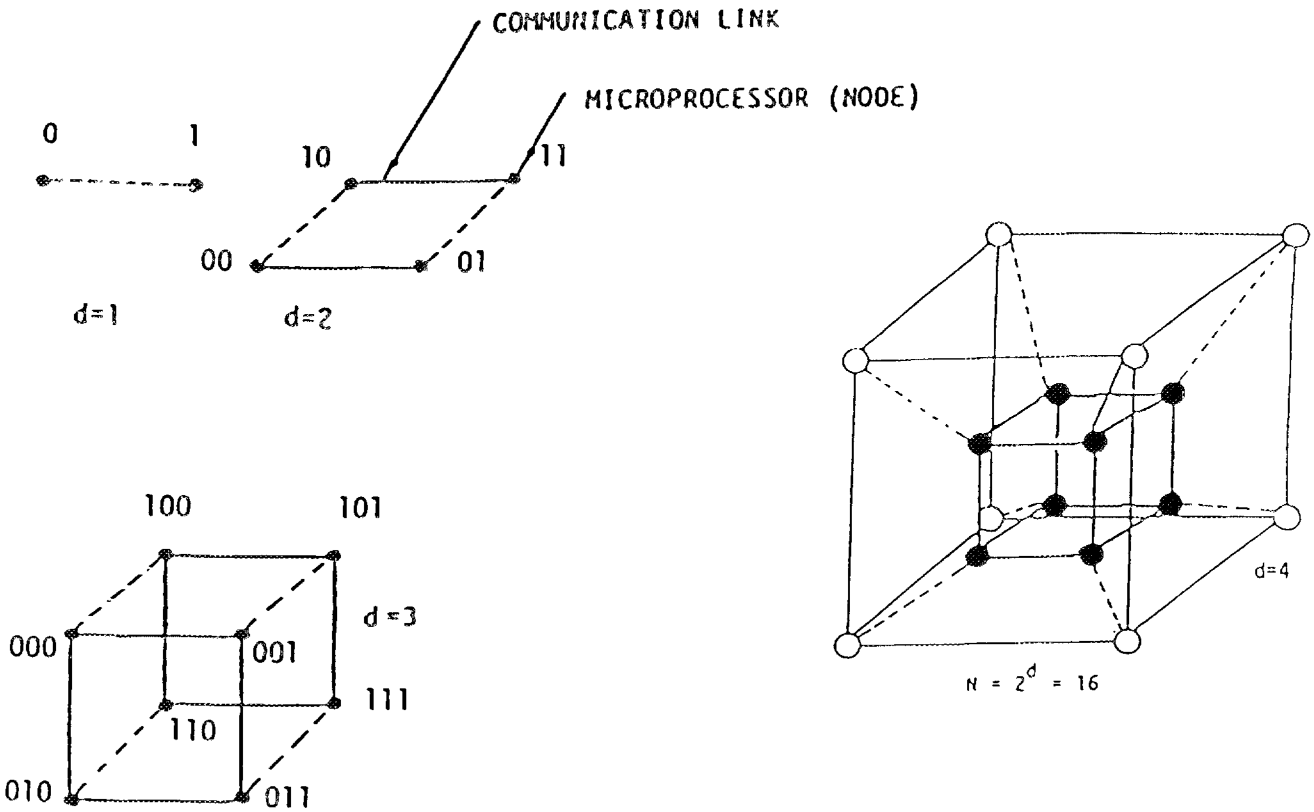

Fig. 1. Hypercube Architecture in d Dimensions. An order-d hypercube is constructed recursively from two order-(d-1) cubes by connecting nodes having a Hamming distance of one, through the most significant bit of their identifying number (dotted lines).

denote nodes and lines refer to communication channels. It is important to notice that hypercubes can be constructed in a modular fashion, i.e., an order-d hypercube is constructed from two order-(d -1) cubes by connecting appropriate nodes.

Several architectural advances of ensemble machines are of special significance to us. Previous multiprocessors[8] were generally constructed to allow direct implementation of conventional programming constructs. In particular, these multiprocessors typically include special switching hardware to allow each processor to access the memory of others. A range of problems are associated with these tightly coupled multiprocessor architectures. Of particular impact is the fact that hardware cost and complexity grow much faster than linearly with the size of the machine, resulting in an ever increasing loss of efficiency of the software as the number of processors is raised (e.g., as in CMU's C.mmp).[9] On the other hand, research at Caltech and elsewhere has indicated that architectures communicating through message passing (e.g., such as hypercube ensembles) have better properties.

Two architectural characteristics make the hypercube ensemble machine particularly attractive for CESAR applications. The first refers to communication time between nodes. For example, consider a 12-dimensional cube ($N=2^{12}=4096$ processors). It is homomorphic to a 64 x 64 square grid. However, the most distant nodes in the latter are 126 channels apart, but only 12 in the former. The second characteristic refers to symmetry. The system looks topologically identical from the point of view of each node there are no corner vs. edge, or root and leaf nodes as in regular grids or trees. This property will simplify the dynamic reallocation of subcubes by ROS, to whatever task requires additional computing power.

b. Developing Algorithms for Ensemble Machines

An essential step to insure the successful implementation of ensemble machines as "brains" of future HERMIES robots (or, e.g., of autonomous land vehicles, futuristic airplanes, space stations, weapon systems,...) is the development of adequate algorithms for concurrent computation. It should be pointed out that this task is far more difficult in the framework of intelligent systems, than for the usually demanding computations encountered in the classical fields of science and engineering.[10] In the latter (including, for example, matrix, grid or finite element formulations[11]) the algorithm structure is so regular, that the corresponding processes (a "process" is simply an instance of a sequential program augmented by message passing primitives, and may represent, for example, computation of an equation term) can be mapped directly onto the hardware topology.

For intelligent machine applications, typical process structures are irregular and also involve nonlocal communications. This requires that an optimal (or near-optimal) mapping of the process structure (task graph) onto the ensemble be computed.[12] Even for "static" process structures (i.e., those with a non time-varying topology) where the mapping can be computed prior to execution, this endeavor is extremely difficult, particularly when precedence constraints are involved. A prototype mapping system, ROSES (ROS Experimental Scheduler) developed for our DOE robotics activities, is currently being tested,[12] and shows excellent promise.[13] An application for which a near-optimal mapping has been achieved is the solution of the inverse dynamics equations.

The situation becomes considerably more complex if the process structure evolves dynamically, as may be required for intelligent machines operating in unstructured environments. Complications include the development of appropriate methodologies for real-time mapping and remapping of task graphs onto the machine's topology, the capability for processes to spawn or annihilate other processes, and most importantly for the operating system to be capable of load-balancing the activities of all processors to achieve optimal utilization and throughput. Our approach is outlined in the following section.

III. INTELLIGENT MACHINE OPERATING SYSTEM

Our intent is to develop essential components of a methodological framework for real-time systems capable of fully exploiting the fundamental computational breakthrough offered by ensemble machines for concurrent computation. A Phase-1 effort attempts to develop dynamic resource allocation (scheduling and load-balancing) algorithms in a virtual time environment. The application areas targeted might include either autonomous robotics, avionics, or SDI tasks, with an initial demonstration limited to a relatively "simple" problem. In the following discussion, in order to fix the ideas, all nomenclature will refer to robotics.

a. Basic Concepts

Each of the many activities taking place in a robotic system (e.g., vision, sensing, manipulation, ...) will be represented by many asynchronous interruptable entities called "processes" or "objects". Processes may be grouped dynamically into "tasks". In particular, device control processes will correspond to, track and control each hardware component.

In the same vein, equation sets (e.g., the inverse dynamics equations) will be partitioned into precedence-constrained processes.

It should be emphasized that it is highly desirable that an ensemble machine be dynamically reconfigurable into a set of ensembles of lower dimensionality. For example, on the ORNL/NCUBE, one 6-dimensional cube is equivalent to eight 3-dimensional cubes, or to one 5-dimensional and two 4-dimensional cubes, etc. The intent is to assign major robot activities to specific partitions. The corresponding processes will be distributed among the processors of each ensemble subset, and should be movable between them at any time to preserve the locality of communication and load balancing. This is also essential to insure sustained system performance when the machine size is scaled up. Obviously, dynamic reconfigurability is desirable both for "coarse" as well as "fine-grained" concurrent computation ensembles.

b. Virtual Time Environment

One of the principal functions of an operating system for an Intelligent Machine is to coordinate processes, the activities of which may refer to times other than real time (i.e., wall-clock time). There are several categories of such "non-real" times, which together will be lumped under the name "Virtual Time".

(1) One category of "non-real" time arises in simulation of the future, a necessary element of planning. When the task of simulation is carried out by a group of asynchronous processes running concurrently on a number of processors, each process will in general be at a different point of the simulation - i.e., at a different "virtual time". It is the

responsibility of the operating system to coordinate the interaction of these processes (via inter-processor messages) in a manner which (a) preserves the logical consistency of the world model, (b) is as efficient as possible, and (c) is transparent to the system's user.

(2) Another category of Virtual Time is future real time. Commands or directives to effectors often must have a real-time dimension. For example, a directive to a robot arm may consist of a trajectory, comprising a series of motions, which are to be coordinated in time with the trajectory of another robot arm. A planning task issues streams of directive messages - and messages changing earlier directives - to the effector tasks. The actions of different effectors must remain coordinated in spite of changes, delays, and so forth. The operating system must provide facilities for the coordination of timing, under these conditions.

(3) Yet another category of Virtual Time arises in calculations (e.g. solving equations) which, in order that results be obtained with sufficient speed, must be carried out asynchronously (i.e, "chaotic relaxation") by a group of concurrent processes.

To implement these concepts, our work builds upon the basic techniques of Time-Warp simulation,[14] and extend them as needed for real-time implementation. The Time-Warp mechanism is used essentially to speed up simulations by solving the problems in (1) above. In a simulation, each process (usually representing a particular individual component of the simulated system) keeps track of its own simulation time (ST). An inter-process message must be stamped with the ST at which its receiver process must act upon it, and each process maintains a queue of its input messages, arranged in order of ST.

Under a Time-Warp based operating system, a process will act at any time upon its input message of lowest ST. If later a message of lower ST arrives, the process "rolls back" to a simulation stage prior to that ST, continues forward again, and issues an "anti-message" for each previously issued message now found to have been incorrect. When a process receives an anti-message to a message not yet acted upon, these two annihilate each other. If the message has been acted upon, the process rolls back to a prior simulation stage. Thus under Time Warp a process may execute whenever a new message arrives and deadlock cannot occur as it does with previous methodologies.[15]

The operating system we are in the process of building will provide facilities for coordinating various types of Virtual Time (VT) including simulation time and future real time. Our current implementation on the enhanced Caltech hypercube simulator includes:

- Facilities for passing messages between processes, queuing input messages to a process in order of VT-stamp, and starting process execution on receipt of messages;
- Time-Warp mechanisms for the simulations required for intelligent planning, including anti-messages, annihilation of queued messages by their anti-messages (and v.v.), and rollback to previous states;
- The use of anti-messages as the means for cancelling future-real-time commands (e.g., trajectories as communicated from planning tasks to effector tasks) which have not yet been acted upon;
- Support for the response of effector tasks to cancelling or changing of previous commands - the difference from simulation rollback being that previous actions in the real world cannot be erased.

Ultimately the IMOS/VT operating system would thus apply and extend Time Warp methods to meet the demands of implementing real-time control with a set of concurrent processes running on a multiprocessing ensemble.

c. Fundamental Design Characteristics of IMOS/VT

As pointed out earlier, there are roughly two classes of applications for concurrent computation ensembles. We shall refer to them in the following as "Class-I and Class-II in the following discussion. Class-I applications, generally representative of the problems encountered in the classical fields of science and engineering are characterized by a very regular (crystalline) structure in space and time that is known statically; their model of communication assumes no loss of information, and they are of computation-intensive nature. Class-II applications tend to have the opposite characteristics, i.e., they are irregular in space and time, with the irregularities sometimes unpredictable in advance; possible loss of information during communication (e.g., loss of some sensor readings to perform an operation at higher rate) may occur, and they are just as often communication-bound as compute-bound.[16] These differences provide the fundamental guidelines for our research and pervade all aspects of the IMOS/VT design.

IMOS Process Management. For Machine Intelligence (i.e., Class-II) applications we do not expect to know in advance the sizes or computational demands of all our tasks, which may expand or contract in numbers of processes depending on the environment. Thus, we need IMOS/VT to provide runtime monitoring and load-balancing of various kinds. In some applications (especially planning) it will be routine to create and destroy tasks and their processes frequently. Processes must be small, and creation and destruction operations within IMOS must be optimized.

IMOS Memory Management. A much more flexible approach is necessary. processes are dynamically expanded, contracted, created, destroyed, duplicated, and moved. Their communication patterns are likely to be irregular and changing, requiring dynamic buffer allocation and flow control.

IMOS Message Communication and Synchronization. We expect very frequent communications, with possibly great irregularity in time, length and space patterns of communication. Hence buffering, flow control and packeting must be provided by IMOS. To achieve the highest possible performance rate, some loss of information during message communication may be unavoidable. Message routing strategies must also be more complex, because of the desire to route around congestion or failed nodes or channels, which would otherwise cause deadlock. Furthermore, if processes are in motion the message routing strategies must handle moving targets.

Synchronization is also difficult to handle, since the receiving process does not necessarily know how many messages to expect (if any!), or from "whom", or when. An approach suggested by Jefferson would require the operating system must cause a "software interrupt" in the receiving process when a message arrives. Finally, if processes can be in motion, order-preserving communication is expected to be costly.

IV. LOAD BALANCING IN HYPERCUBE MULTIPROCESSORS

Load balancing algorithms are required for dealing explicitly with the allocation of resources in a concurrent computation ensemble. The goal is to minimize execution time by evenly distributing the task loads across the system, while minimizing interprocessor communication. The difficulty in solving this problem lies in the conflict of constraints over a configuration space which grows exponentially with the number of tasks. In particular, the goal of minimizing interprocessor communication, to avoid saturation-effect bottlenecks which degrade performance, requires that tasks be "clustered" on few, adjacent nodes; on the other hand, to even the processor loads requires that tasks be spread out over all nodes.

The load balancing problem is closely related to multiprocessor scheduling, a subject matter which has been studied extensively over the past twenty years, and for which excellent reviews can be found in the literature. Major difficulties arise when the number of tasks required by a particular algorithm exceeds the number of available processors, and/or when the interconnection topology of the task graph, as obtained from the precedence constraints, differs from the interconnection topology of the computation ensemble. Optimal schedules are in general extremely difficult, if not impossible to obtain, since for an arbitrary number of processors, unequal task processing times and non-trivial precedence constraints the problem is known to be NP-complete.

1. Basic Concepts in Load Balancing

Static load balancing methods permanently assign newly created processes to what appear at that moment to be the best nodes. These processes are not moved once their execution is initiated, under the assumption that their runtime characteristics do not later change in such a way as to cause nodes to become very unbalanced. Load balancing thus occurs only when a new process is created. For precedence constrained tasks this represents the current state of the art.

To adapt to potential changes in the runtime characteristics of processes, one needs to develop "dynamic" load balancing algorithms. Such algorithms may require that processes be migrated during their lifetime to better nodes to provide much needed efficiency, particularly for a large ensemble that shares multiple activities. Load balancing would occur at any time, rather than being limited to times when new processes are created.

To address the load balancing problem, we are currently exploring the applicability of the simulated annealing method.[17]

2. Simulated Annealing

Kirkpatrick et al. have pointed out the analogy between the behavior of condensed matter at low temperatures and combinatorial optimization problems.[17] They proposed a new optimization methodology, referred to as "simulated annealing", which uses techniques suggested by statistical mechanics to find global optima of systems with large numbers of degrees of freedom. The simulated annealing algorithm can be sketched as follows. Consider a combinatorial optimization problem specified by a finite set C of configurations (or states) X, and by an objective function E defined over X. From equilibrium statistical mechanics we know that all configurations $X = (x_1, \ldots, x_N)$ are possible, but that the probability of observing a given X is governed by the canonical distribution:

$$p(X) = \frac{\exp[-E(X)/\theta]}{\sum_{X \in C} \exp[-E(X)/\theta]}$$

Here θ refers to the product kT of the Boltzmann constant by the absolute temperature for a physical system, and will represent a control parameter ["effective temperature"] in the optimization analogue. The problem is then to find the configuration X which induces the minimum value of E.

The algorithm starts from an initial state X and follows a sequence of annealing temperatures θ_0, $\theta_1,\ldots,\theta_i,\ldots$ where $\theta_{i+1} < \theta_i$. The algorithm can be summarized as follows:

[1.0] loop over temperature index i

  [2.0] set $\theta=\theta_i$

  [2.1] loop over sample size at temperature θ

    [3.0] generate new state $X' = F(X)$ where
      F represents a heuristic that tends to
      select states with lower E

    [3.1] $\Delta E = E(X')-E(X)$

    [3.2] If $\Delta E < 0$ then
        accept new configuration unconditionally
        i.e. $X=X'$
      Else
        accept new configuration only if it satisfies
        the Metropolis Criterion, i.e.
        r = uniform-random (0,1)
        If $r < \exp[-\Delta E/\theta]$ then $X=X'$
      End if

  [2.2] end loop over sample size at temperature θ

  [2.3] compute average $\langle E\rangle_i$ at temperature θ

  [2.4] If $(\langle E\rangle_i-\langle E\rangle_{i-1})/\langle E\rangle_i < \varepsilon$ then display results and stop

[1.1] end loop over temperature index i.

As noted already, an essential feature of the Metropolis procedure is the possibility to include states which increase the value of the objective function. This allows eventual escape from local minima of E in the configuration space, thus reducing the chances of entrapment in a suboptimal solution. Current areas of active research address the development of methods for effective selection of new configurations [i.e., selection of the function F], as well as the determination of appropriate annealing schedules [i.e., selection of annealing temperatures θ_i and sample sizes at these temperatures].

3. Implementation Approach

We are exploring both static and dynamic load balancing. The implementation is being carried out within the framework of the ROSES system[12]. The current version of ROSES was developed to provide uniquely powerful scheduling capabilities for mapping precedence-constrained task graphs onto a concurrent computation ensemble. Although this problem is NP-complete, ROSES achieves near-optimal solutions by combining heuristic techniques for handling time complexity with special instances of abstract data structures to handle space complexity. Currently ROSES assumes a non-preemptive scheduling approach: whenever there is a processor ready to be assigned a task, an individual assignment is made. Each assignment corresponds to a "base point", i.e., one may vary the scheduling solution only by changing each individual assignment, while the time point and processor under consideration remain unaltered. At each base point, all tasks ready to be assigned (because their precedence requisites are satisfied) constitute a "set of alternatives" (or A-set). The A-set is constructed and updated in such a way as to continually satisfy the precedence constraints. Choosing a process (i.e., task) for execution from an A-set is guided by heuristics combined with graph-theoretic impasse detection techniques.

ROSES is being modified so that static load balancing would involve the implementation of the simulated annealing algorithm at the A-set level. The modified ROSES kernel would then be run on the NCUBE controller board, prior to task execution on the nodes.

To implement dynamic load balancing we need to significantly extend the ROSES methodology. In particular, we need to allow for task preemption, and to provide additional support by developing three classes of algorithms to be implemented by the operating system resident on each node of the hypercube. The first class will contain information exchange algorithms, to be responsible for the continuous exchange of load information and "task bidding" data between the processors. The second class of algorithms will be used by each processor to monitor its own load on a continuous basis in order to determine whether it can guarantee the execution of newly arriving tasks, or whether such tasks should be migrated. The third class of algorithms will handle process migration; mechanisms need to be implemented to move both code and data, and to reroute the logical communication paths. This should provide a significant measure of dynamic balancing on a short range, fast response scale. ROSES, at a higher hierarchical level, would then attempt to drive the system to global equilibrium, by applying generalized simulated annealing beyond the A-set level.

V. SUMMARY AND FUTURE DEVELOPMENTS

Phase-1 of our project corresponds to a "simulation" stage of the IMOS/VT development. During this period we expect to complete the design and development of the following items:

I_1: Extend ROSES methodology[12] to enable handling of dynamically evolving systems (currently ROSES can map only static task graphs onto concurrent computation ensembles); specifically, allow for preemptive scheduling and add algorithms for

- Load information exchange among subensembles;
- Measuring a processor load
- Process migration.

I_2: Develop algorithms for global resource allocation using an appropriately modified simulated annealing approach.

I_3: Develop a Virtual-Time/Real-Time kernel, to control execution of processes on a distributed system through resolution of inter-process conflicts in virtual time and under real-time constraints; include handling of inter-process message queuing and preemption by emergency messages.

I_4: Implement items I_1-I_3 initially as an IMOS/VT simulator. The simulator will be written both in C.

Our long-range objective is the design of a complete Intelligent Machine Operating System (IMOS/VT) and its implementation on an advanced VLSI-based multiprocessor system. For experimentation purposes on the advanced NCUBE hardware, a "prototype" version of the system will be developed in a Phase-2 effort. The "prototype" version of IMOS/VT would include enhanced capabilities for all items developed under Phase-1. In addition a lowest level hardware interface will be implemented to provide services including loading of processes, handling interprocessor I/O, an asynchronous broadcast facility, real-time clock handler, interrupt and trap handlers for I/O, timeout, runtime errors, etc. The prototype version will also provide application-oriented development tools.

Specifically, we conceive that the application programs of interest would run above IMOS/VT as a set of asynchronous tasks, including:

- Goal-directed planners, (e.g., for manipulator motion, platform motion,...) incorporating simulators of possible future scenarios of action, and decision mechanisms;
- Various calculation tasks, such as that for robot dynamics and control;
- Hierarchical navigation tasks, such as those for carrying out complex trajectories with avoidance of moving obstacles;
- Hierarchical sensing tasks, including vision and sensor fusion.

In summary, the next phases of the project will be concerned with the full implementation of IMOS/VT on the NCUBE concurrent computation ensemble, and its application to real, complex problems of interest to CESAR.

REFERENCES

1. C. R. Weisbin, J. Barhen, G. de Saussure et al., "Machine Intelligence for Robotic Applications," Procs., Conference on Intelligent Systems and Machines, Oakland University, Rochester, MI (April 1985).

2. R. B. McGhee, D. Orin, D. R. Pugh et al., "A Hierarchically Structured System for Computer Control of a Hexapod Walking Machine," 5th IFTOMM Symposium on Robot and Manipulator Systems, Udine, Italy (June 1984).

3. Defense Advanced Research Project Agency, "Strategic Computing," U.S. Department of Defense (October 1983).

4. J. Barhen, C. R. Weisbin and G. de Saussure, "Real-Time Planning by an Intelligent Robot," Procs., 1984 International Computers in Engineering Conference, pp. 358-360 Las Vegas, NE (August 1984).

5. C. L. Seitz, "Concurrent VLSI Architectures," IEEE Trans. Comp., C-33, 1247 (1984).

6. C. L. Seitz, "The Cosmic Cube," Comm. ACM, 28, #1, 22 (1985).

7. P. B. Schenk et al., "Parallel Processor Programs in the Federal Government," IEEE Computer; 18, #6, 43 (1985).

8. R. H. Kuhn and D. A. Padua, Parallel Processing, IEEE Computer Society Press (1981).

9. R. D. Fennell and V. R. Lesser, "Parallelism in Artificial Intelligence Problem Solving: A Case Study of Hearsay-II," IEEE Trans. Comp., C-26, 98 (1977).

10. G. C. Fox and S. W. Otto, "Algorithms for Concurrent Processors," Physics Today (May 1984).

11. G. A. Lyzenga, A. Raefsky and G. H. Hager, "Finite Elements and the Method of Conjugate Gradients on a Concurrent Processor," Proc., 1985 International Computers in Engineering Conference, pp. 401-405, Boston, MA (August 1985).

12. J. Barhen, "Robot Inverse Dynamics on a Concurrent Computation Ensemble," Procs., 1985 International Computers in Engineering Conference, pp. 415-429, Boston, MA (August 1985).

13. J. Barhen, "Hypercube Ensembles: An Architecture and Algorithms for Intelligent Robots," book chapter in J. Graham, ed., Special Computer Architectures for Robotics, Gordon & Breach, NY (Winter 1985).

REFERENCES (cont'd)

14. D. Jefferson and H. Sowizral, "Fast Concurrent Simulation Using the Time-Warp Mechanism," Rand Corporation (June 1983).

15. K. M. Chandy and J. Misra, "Asynchronous Distributed Simulation via a Sequence of Parallel Computations," Comm. ACM, 24, #11, 198 (1981).

16. D. Jefferson, presentation to CESAR (June 1985).

17. S. Kirkpatrick, C. Gelatt and M. Vecchi, "Optimization by Simulated Annealing," Science 220, 671 (1983).

Paper 4

PLANNING FOR TRANSIT IN UNKNOWN NATURAL TERRAIN

S. Y. Harmon

Code S442, Naval Ocean Systems Center
San Diego, CA 92152

ABSTRACT

The Ground Surveillance Robot (GSR) development has reached a stage where a route planning system is potentially useful. The GSR's route planning problem can be solved by having a generalized planning engine operate upon a data base containing orienteering knowledge and models of the vehicle and the surrounding environment. Orienteering knowledge contains heuristics for position finding, route planning and route following in undeveloped terrain. Some orienteering hints valuable to an autonomous vehicle planning transit through unknown undeveloped territory are discussed. In addition, the requirements for a sufficient planning mechanism to support autonomous transit through unknown natural terrain are also discussed.

INTRODUCTION

Some years ago a project began to implement an autonomous vehicle demonstration, called the Ground Surveillance Robot (GSR), for a practical environment [1]. Since that time implementation of the numerous sensor and control subsystems, which are necessary parts of such a vehicle, has proceeded. From the beginning of that project the question "How will such a vehicle plan its route through unknown natural terrain?" has been asked. For years the GSR researchers have been too mired in the details of low level implementations to do any more than define the present limitations in planning systems for unknown natural terrain [2]. This is not to imply that over these years the route planning question has been neglected. While this paper will not answer the route planning question completely it will address the issue and reflect a current state of thought slightly tainted by the realities of implementation.

The GSR's goal is to transit to a known location over unknown natural terrain. The present architecture provides a uniform view of highly processed sensor information through a blackboard which represents the best assessment of the world condition at any time [3]. This assessment is built from numerous sometimes overlapping sensor sources. The blackboard represents sensor information on: the relative location of nearby (< 10 m) obstacles/hazards, rough absolute geographic position (w/i 100 m, every 90 min), vehicle speed, heading, roll angle and pitch angle (w/i 0.05 m/s & 1.0 deg) and an uncertain, incomplete and inaccurate terrain model containing information on obstacle statistics, terrain variability and terrain surface nature. The blackboard also contains relative location estimates of local terrain features for relative navigation. The bulk of the terrain information is constructed from high resolution gray level imagery, low resolution color imagery & point range estimates collected from cameras and a computer controlled laser range finder mounted on a computer controlled transport system [4].

Needless to say numerous authors have discussed the route planning issue and related topics. Many of the techniques which have been developed deal with space in terms of free space and obstacles where space can be traversed and obstacles cannot [5-16]. For the most part, these techniques provide solutions to the obstacle avoidance problem which is extremely useful in a structured well mapped environment such as a modern automated factory. However, the GSR philosophy is that this problem should be handled at a low level by the locomotion control system using potential field avoidance techniques similar to those discussed in [17,18]. While this approach does not guarantee an optimal path for short range travel it does free the limited onboard computational resources for higher level planning activity to optimize for long range travel. In the GSR, the division between low level route planning and high level route planning occurs where the simple free space/obstacle representation fails and detailed terrain modelling is necessary. Similar concepts have been suggested in the form of hierarchical planners [19-22]. However, the GSR approach is the simplest of all those described and distributes planning activity to lower subsystems more than others. A limited number of techniques reduce space to go and no-go regions on a large scale for outdoor domains [5,14,16,23,24]. However, all but one of these uses a map provided before the journey begins. The GSR has no a priori map.

Most of the existing route planning efforts use some form of tree or graph representation of traversible space and they search the graph using cost functions and heuristics for optimization [5,6,8-11,13-16,19,21,22,24]. Other approaches include formulating the problem as an optimal control problem and minimizing a cost function described in state space representation [12], a wandering standpoint algorithm to search space [20] and a combination of a script based problem solver, special purpose algorithmic problem solvers and domain specific production rules [23]. Most of these problem solving techniques are tailored quite specifically to route search and, therefore, offer no solution to other problems related to finding one's way in the great unknown such as how to collect sufficient information from the surroundings to make intelligent decisions about which way to proceed. In fact the planning of activities solely to collect information has been described as a special case of problem solving for which there is likely no general purpose algorithm [21].

The global planning problem, where obstacle avoidance is ignored, can be described as planning a connected series of paths to a point at the fringe of sensor coverage in the direction of the goal location considering the relations between the environmental and vehicle constraints. This statement assumes that the vehicle always proceeds toward the goal to the frontier of its sensing ability, a technique used by others with mobile robots planning transit in unknown areas [7,8,16]. Environmental constraints are imposed by slope, terrain variability, obstacle density, ground cover and terrain surface composition. Vehicle constraints include limited ability to deal with slopes, finite vehicle size, minimum required surface normal and shear strengths, finite fuel capacity, load and time dependent fuel consumption and finite sensor range and field of view. Most of the references cited above approach some aspect of this overall problem. However, in the GSR the global path planning problem is seen in a more general way. The planning mechanism is decoupled from the domain. Decoupling the domain independent activity from the domain dependent activity is not new but has not been done in any of the techniques discussed above. Even the generic graph search techniques couple their planning mechanism to the domain through embedded cost functions and heuristics.

Decoupling the global route planning problem produces two subproblems, orienteering and planning. Orienteering is the human skill of finding a route through unknown natural terrain. This skill has been practiced for thousands of years by humans and its principles are discussed in several sources [25-27].

On the other hand, planning is a much more difficult problem. Generalized planning mechanisms are very poorly developed. Although some progress has been made in the areas of evidential and temporal planning many capabilities demanded by route planning in a complex environment are as yet unavailable in a single mechanism. One solution to this problem is to develop a planner composed of several different planning mechanisms each appropriate for different special situations [16,23]. While this solution addresses the problem of generality it does so by increasing the complexity of the planning mechanism rather than of the data base upon which the planner operates. It also introduces the problem of meta-planning, a much more poorly understood problem than just planning for a specific domain. Clearly, a single general planning engine able to perform route planning for an autonomous mobile robot would be the best solution.

The remainder of this paper discusses some orienteering principles relevant to autonomous vehicle transit and some of the requirements for a planning engine ideally suited for the route planning problem executed in an autonomous robot.

Orienteering

Orienteering is the art of path finding through unknown territory [25]. This knowledge, like other forms of human knowledge, is contained by various sources. The knowledge is strongly environmentally dependent so path finding knowledge varies with different terrain type. Models of various elements of the environment (e.g., terrain structure, vegetation coverage, weather, celestial objects) are necessary for successful orienteering. Orienteering knowledge consists of information for direction finding, route planning and route following techniques. Examples of this information are presented briefly below.

Direction finding. Several reliable techniques exist for finding absolute direction in unknown territory. The most reliable and unambiguous techniques rely upon celestial objects but there are other techniques which do not. The North Star and the Southern Cross can provide a reasonably accurate notion of absolute direction. The directions of rising or setting Sun or Moon also provide absolute direction [26,27]. The direction of shadow movement produced by the light from either the Sun or Moon indicates the west to east directions during most of the daytime when other celestial clues are not available [25,26]. However, all of these techniques assume that the sky and celestial objects are visually accessible. In travel in deep canyons or under a foliage canopy the sky is largely unavailable and other techniques must be used.

Fortunately, many natural environmental signs can provide direction information. By knowing the direction of the prevailing winds in a geographic area one can find direction from drifted snow, sand or dust rippling, leave shedding and tree deformation. The thickest moss grows on the seasonally coolest side of trees and, for solitary trees, this is the north side. Also flowers face toward the south to receive the most available sunlight [27]. These and other environmental signs guarantee knowledge of global orientation. This knowledge is critical for path finding toward a known goal location.

Route planning. The process of route planning without a map involves collecting information about the surroundings using sensor resources and using that information to decide the most effective route to take to the goal. Starting with absolutely no information about the local terrain the robot must plan actions to collect sufficient terrain knowledge to begin successful transit. However, if too much time is spent building the environmental model then the vehicle will not be able to reach the goal. Thus, there must be a balance between time spent collecting terrain information and time spent approaching the goal location. The time necessary to collect terrain information depends heavily upon the complexity of this terrain. In flat desert country the robot could proceed directly toward the goal with little or no additional terrain information other than that collected during the journey. In more complex terrain the robot may first need to proceed to the highest local terrain feature to survey the surroundings. At that time, the existence and relative locations of major terrain features should be noted including mountain peaks, rivers, lakes, large open areas, valleys and canyons [26]. This provides information about good and bad paths as well as landmarks for relative navigation.

Several heuristics have been developed to aid planning a route through undeveloped territory. For instance, traveling on ridges and divides is easier than traveling in valleys or along streams. The vegetation is often less dense; the outlook is better for landmark navigation and there are fewer streams and swamps to cross [26]. Following streams is difficult because of fording, detours and thick vegetation. In mountainous terrain the falls, cliffs and side canyons associated with rivers and streams create significant difficulties following the desired route. In flat country, streams meander; vegetation is dense; swamps are common and outlooks are rare. Advantages in following streams are that they may lead to inhabited areas and they give unmistakable reference in strange country [26]. If streams must be crossed the current is slowest and the depth is shallowest at the widest places [26]. Also, the inside bends provide shallows for exiting the stream (noting that it is often easier to get into a stream than to get out of it). If swamps are encountered emergent vegetation usually indicates more supportive ground than areas of mud or open water [26]. Mountainous areas present special difficulties for path planning. The many ridges and valleys in such terrain significantly increase planning complexity and present problems for path following and position finding. In these areas, following ridges or valleys is often easier than crossing them [26].

Route following. Once a route has been planned, following that route through complex terrain may be difficult, especially if it is covered with thick vegetation, because of the many detours forced by untraversible environmental features. As a result, several tricks have been developed to follow a desired route. Straight line travel is possible by traveling on the line formed by three landmarks (i.e., one behind and two in front). When a landmark is passed another on the desired course is chosen. This technique provides an easy way to visually follow a straight line even in the most complex terrain [27]. Triangulating from local landmarks can be used to identify points at which course changes are necessary. The landmarks should be outstanding features such as mountains, ridges, drainage patterns and uniquely distinguishable vegetation [26]. These landmarks can be used to relocate a desired path if a detour is necessary due to some untraversible environmental feature [27]. At night fixing on a single star in the direction of desired travel will also assist holding a constant course [26].

Planning

Unfortunately, the heuristics described above are not sufficient in themselves to accomplish global route planning. An underlying planning mechanism is required. Several generic planning mechanisms have been proposed but few of them have been implemented for mobile robot purposes. Several requirements for such a planning engine can be formulated.

The desired mechanism must plan in time over widely different time scales. It should be able to respond in seconds to an unexpected emergency and be able to relate events occurring over several hours. The desired planning engine must simultaneously model many different environmental conditions and respond to a wide variety of circumstances. The planner must handle uncertain and incomplete information which is received incrementally over time from several different sensor sources. This implies that the planning must be a continuous process of creating expectations of future situations and corresponding action plans to manipulate those situations toward the desired goals. Planning with both goals and constraints should be possible. Since the robot can change its position without the planner's help the planner needs to synchronize its activity with real time occurrences. This is not to imply that the planner must be a real time planner since, in the GSR, much of the obstacle avoidance planning load is supported by the locomotion processor which runs in parallel with the global planner. This organization relieves the planner of the need to respond quickly if the vehicle absent mindedly stumbles into some hazard. In the GSR, these circumstances are safely stabilized by the locomotion controller before the planner is needed. As a result, the GSR planner must only be able to respond within a few minutes for most situations but it must keep pace with the evolution of those situations so as to prevent getting hopelessly behind. Ideally, the planning engine will also be independent of the application specific data base and it should be a uniform and simple mechanism which operates upon a large domain dependent data base.

CONCLUSIONS

This paper has discussed breaking the route planning problem in unknown natural terrain into a domain specific part and a domain independent part. The domain specific knowledge can be derived from orienteering information sources. The domain independent part is planning. Several requirements for a sufficient planning mechanism have been presented. Today no sufficient general purpose planning engine has been introduced which meets all of these criteria although several mechanisms have been developed which have approached subsets of these requirements.

REFERENCES

[1] S. Y. Harmon & M. R. Solorzano, "Information Processing System Architecture for an Autonomous Robot System", Oakland University Conf. on Artificial Intelligence, Rochester, MI, 26-27 Apr. 1983, p1-11.

[2] S. Y. Harmon, "Comments on Automated Route Planning in Unknown Natural Terrain", IEEE Conf. on Robotics, Atlanta, GA, 13-15 Mar. 1983, p571-573.

[3] S. Y. Harmon, "Coordination of Intelligent Subsystems in Complex Robots", IEEE Conf. on Artificial Intelligence Applications, Denver, CO, 5-7 Dec. 1984, p64-69.

[4] S. Y. Harmon, "USMC Ground Surveillance Robot (GSR): A Testbed for Autonomous Vehicle Research", 4th Univ. of Alabama, Huntsville Robotics Conf., Huntsville, AL, 24-26 Apr. 1984.

[5] F. P. Andresen et al., "Visual Algorithms for Autonomous Navigation", IEEE Conf. on Robotics & Automation, St. Louis, MO, 25-28 Mar. 1985, p856-861.

[6] V. D. Belenkov et al., "Adaptive System for Control of Autonomous Mobile Robot", Engineering Cybernetics, 16(6), Nov/Dec 1978, p37-45.

[7] D. F. Cahn & S. R. Phillips, "ROBNAV: A Range-Based Robot Navigation and Obstacle Avoidance Algorithm", IEEE Trans. on Systems, Man & Cybernetics, Sept 1975, p544-551.

[8] R. Chatila & J. P. Laumond, "Position Referencing and Consistent World Modeling for Mobile Robots", IEEE Conf. on Robotics & Automation, St. Louis, MO, 25-28 Mar. 1985, p138-145.

[9] J. L. Crowley, "Navigation for an Intelligent Mobile Robot", IEEE Journ. of Robotics & Automation, RA-1(1), Mar. 1985, p31-41.

[10] G. Giralt et al., "A Multi-Level Planning and Navigation System for a Mobile Robot: A First Approach to HILARE", 6th Int. Joint Conf. on Artificial Intelligence, Tokyo, Japan, 20-23 Aug. 1979, p335-337.

[11] Y. Ichikawa & M. Senoh, "A Heuristic Guidance for Automated Vehicles", IEEE IECON '84, Tokyo, Japan, 22-26 Oct. 1984, p313-317.

[12] W. G. Keckler & R. E. Larson, "Control of a Robot in a Partially Unknown Environment", Automatica, 6(3), May 1970, p469-476.

[13] H. P. Moravec, "The Stanford Cart and The CMU Rover", Proc. of the IEEE, 71(7), Jul. 1983, p872-884.

[14] V. P. Pyatkin & V. Ya. Sirotenko, "Path Planning by Robot", Engineering Cybernetics, 16(6), Nov/Dec 1978, p54-59.

[15] R. Ruff & N. Ahuja, "Path Planning in a Three Dimensional Environment", 7th IEEE Conf. on Pattern Recognition, Montreal, Canada, 30 Jul.- 2 Aug. 1984, p188-191.

[16] A. M. Thompson, "The Navigation System of the JPL Robot", 5th Int. Joint Conf. on Artificial Intelligence, Cambridge, MA, 22-25 Aug. 1977, p749-757.

[17] O. Khatib, "Real Time Obstacle Avoidance for Manipulators and Mobile Robots", IEEE Conf. on Robotics & Automation, St. Louis, MO, 25-28 Mar. 1985, p500-505.

[18] B. H. Krogh, "A Generalized Potential Field Approach to Obstacle Avoidance Control", paper MS84-484, SME Robotics Research Conf., Bethlehem, PA, 14-16 Aug. 1984.

[19] Y. Ichikawa & N. Ozaki, "A Heuristic Planner and an Executive for Mobile Robot Control", IEEE Trans. on Systems, Man & Cybernetics, SMC-15(4), July/Aug. 1985, p558-563.

[20] D. M . Keirsey et al., "Algorithm of Navigation for a Mobile Robot", IEEE Conf. on Robotics, Atlanta, GA, Mar 1984, p574-583.

[21] D. McDermott & E. Davis, "Planning Routes through Uncertain Territory", Artif. Intell. 22(2), Mar. 1984, p107-156.

[22] A. M. Parodi, "A Route Planning System for an Autonomous Vehicle", 1st IEEE Conf. on Artificial Intelligence Applications, Denver, CO, 5-7 Dec. 1984, p51-56.

[23] B. Bullock et al., "Autonomous Vehicle Control: An Overview of the Hughes Project", IEEE Trends & Applications 1983, Gaithersburg, MD, 25-26 May 1983, p12-17.

[24] D. T. Kuan et al., "Automatic Path Planning for a Mobile Robot Using a Mixed Representation of Free Space", 1st IEEE Conf. on Artificial Intelligence Applications, Denver, CO, 5-7 Dec. 1984, p70-74.

[25] W. Hillcourt, Official Boy Scout Handbook, Boy Scouts of America, Irving, TX, 1979.

[26] F. C. Craighead, Jr. & J. J. Craighead, How to Survive on Land & Sea, U. S. Naval Institute, Annapolis, MD, 1953.

[27] C. Ormond, Complete Book of Outdoor Lore, Harper & Row, New York, NY, 1976.

Paper 5

O-THEORY - A HYBRID UNCERTAINTY THEORY

E. M. OBLOW

ABSTRACT

A hybrid uncertainty theory is developed to bridge the gap between fuzzy set theory and Bayesian inference theory. Its basis is the Dempster-Shafer formalism (a probability-like, set-theoretic approach), which is extended and expanded upon so as to include a complete set of basic operations for manipulating uncertainties in approximate reasoning. The new theory, operator-belief theory (OT), retains the probabilistic flavor of Bayesian inference but includes the potential for defining a wider range of operators like those found in fuzzy set theory.

The basic operations defined for OT in this paper include those for: dominance and order, union, intersection, complement and general mappings. A formal relationship between the membership function in fuzzy set theory and the upper probability function in the Dempster-Shafer formalism is also developed. Several sample problems in logical inference are worked out to illustrate the results derived from this new approach as well as to compare them with the other theories currently being used. A general method of extending the theory using the historical development of fuzzy set theory as an example is suggested.

1) Introduction

The problem of dealing with uncertainty in inference and reasoning processes is a complex and difficult one. The information available for reasoning is often uncertain, imprecise, and even vague. Approximate means of dealing with the propagation of such data through inference models is crucial to the success of any machine intelligence program. Although no complete solution to this problem is at hand, several different approaches have been pursued. The use of classical Bayesian inference theory (BIT) for instance, is one such approach to this problem which has yielded some success[1,2]. More recently, several attempts derived from set-theoretic formalisms have provided other insights into its solution. These latter methods are represented by Zadeh's fuzzy set theory[3] (FST) and Dempster-Shafer belief theory[4-6] (DST). An excellent unified review of all three of these theories is presented by Prade[7]. Suitable background material for this paper can be found in this latter review article and the extensive list of references cited therein. Preliminary presentations of this paper also suitable for background material appear in Weisbin[8].

In the present article, a different approach to the uncertainty problem will be developed. The motivation behind attempting to develop another approach in this area can better be understood by taking a closer look at the strengths and weaknesses of each of the three uncertainty methodologies just mentioned. For instance, BIT has a strong, well established probability basis but is weak in its applicability to problems which are formulated in set-theoretic language. FST, on the other hand, has a strong and highly developed set theory background but its basic membership function and set operators are less intuitive and physical than those using probability concepts. In between these two extremes lies DST, which has both a probability and set theory basis. Its strongest point is the capability of representing such concepts as noncommital and vacuous belief. On the other hand, however, it lacks the extensive mathematical developments necessary for more general applicability. In addition to these observations, it should be noted that each of these theories gives quantitatively different results in application. Bayes' theorem[1,4], Dempsters' combination rule[4] and fuzzy set rules for union and intersection[3], the basic laws of BIT, DST and FST, respectively, all are quantitatively different ways of combining uncertainty information. Only in certain limiting circumstances do their results all tend to converge[4,7].

In practice then, it should be clear that all three theories (and possibly some new ones) will probably find extensive use in solving the inference-uncertainty problem. Satisfaction with inferential results, computational efficiency, and ease of representation will be the final measures of success of any of these methodologies in any given problem area.

In this paper, an attempt will be made to bridge the gap between FST and BIT using the set-theoretic strengths of the former and the probability basis of the latter. The focal point of this new approach will be the DST, which will be extended so that additional mathematical operators can be used to propagate uncertainties in a wider range of problems. This hopefully will eliminate one of DST's perceived weaknesses compared with FST. The overall approach taken to achieve this synthesis will be to use the available mathematical developments in FST as a model for extending DST while retaining the probabilistic flavor it has in common with BIT. The resulting theory proposed is, therefore, a hybrid-uncertainty theory using the strengths of both FST and BIT on an enhanced DST base.

2) Basis for O-Theory

The basic starting point for the development of operator-belief theory (OT) is DST. A brief outline of the primitive concepts of this theory needed for OT are given here. The reader is referred to Shafer[4] for more details. To begin, use is made of the set of possibilities Θ, with elements $x_i \in \Theta$, and its power set 2^Θ, with elements $x \in 2^\Theta$. As in DST, a basic probability mass, $m_\Theta(x)$, is assigned to each $x \in 2^\Theta$, with the function $m: 2^\Theta \rightarrow [0,1]$, which is normalized by

$$\sum_{x \in 2^\Theta} m_\Theta(x) = 1 \; . \tag{1}$$

Here, x is a set which is a subset of θ (i.e. $x \subset \theta$) and is, therefore, also an element of 2^{θ} (i.e. $x \in 2^{\theta}$).

This normalized mass distribution defined on 2^{θ} is the 'uncertainty' or 'belief' representation of θ, which will be denoted by $\underline{\theta}$ and referred to as the 'belief set of θ'. Any proper subset of θ which will have a mass distribution assigned to it will be denoted by an underlined capital letter (e.g. the set $A \subset \theta$ with mass assignment $m_A(x)$ will be denoted by $\underline{A}$). The set θ will generally be used to represent the largest, finite possibility set under consideration and, therefore, $\underline{\theta}$ will represent the largest belief set.

At this point it should be noted that the normalization given in Eq.(1) represents a departure from DST, in that the assignment of zero mass to the null element of 2^{θ} (i.e. $m(\emptyset)=0$) is not taken to be such by definition. Mass can be assigned directly to $\emptyset$ or it can be acquired as a result of operations on the set θ. This modification is proposed to allow for the possibility that the original set θ might not have been a complete enumeration of all the possible states of the system under investigation to which mass could be assigned. It also allows $\emptyset$ to be an element of 2^{θ} into which conflicting information can be gathered to represent incompleteness in θ.

Note here, that the use of a normalized mass distribution for assignment of mass to elements of the possibility set 2^{θ} as given in DST, represents an extension of the concepts of probability theory, and OT, therefore, has this extended probability basis as well. This normalization can also be interpreted in a set theory context in a fashion which ties it to FST. That is, the normalization represents the maximum effective cardinality of the possibility set θ, which is unity. If masses are treated like membership functions, this means that at most only one member (and possibly none if $m_{\theta}(\emptyset)=1$) can be the true possibility set member.

Two other constructs from DST will also be used in OT. These are the upper and lower probabilities denoted here as P and B, respectively. Their definitions[4], slightly modified, are given for $\forall x, x' \in 2^{\theta}$, as follows:

$$B(x) = \begin{cases} \sum_{x' \subset x \neq \emptyset} m_{\theta}(x') & x \neq \emptyset \\ m_{\theta}(\emptyset) & x = \emptyset \end{cases} \quad (2)$$

$$P(x) = 1 - B(\tilde{x}) , \quad (3)$$

where $\tilde{x}$ is the complement of x in θ and because of the normalization condition given in Eq.(1), we see that $B(\theta)=1-m(\emptyset)$, another departure from DST. In this new form we also see that, in addition to its original DST interpretations, $B(\theta)$ can now be used as a measure of the effective cardinality of $\underline{\theta}$.

With these definitions, the basic strengths of DST, in being able to assign mass to any element of the power set of 2^{θ} and the ability to have an amount of belief remain uncommited to any particular element of 2^{θ} (i.e. $P(x)-B(x)\geq 0$), are therefore retained in OT.

Further developments of this new theory will now be made by extending this basic framework using analogies derived from the mathematical operators and structures available in FST. In particular a basic set of algebraic operators like the union, intersection, and complement will be proposed first, structural relationships will follow, so that order, dominance and equality can be defined, and finally, a norm will be introduced.

3) Structural Relations and Norm

In order to develop the tools necessary for comparing the uncertainties in various possibility sets, some dominance, order and size relations must first be established. For the order and dominance relations, this is not as easy a task to accomplish as it was for FST. That is, the analogy to set inclusion can not be used in OT, since the probability masses represent a distribution over the power set 2^{Θ} and one normalized distribution is not easily included in another. In this case then, the concept of the moment of the distribution was used instead to define an order.

Defining the cardinality of a set x to be |x| (i.e. the number of elements in the set), the dominance of any one member x of a belief set $\underline{\Theta}$ over any other member x' can be defined in terms of a cardinality moment as

$$x \succsim x' , \quad \text{if} \quad m_{\Theta}(x)|x| \geq m_{\Theta}(x')|x'| , \qquad (4)$$

where $\succ$ represents dominance, $x, x' \in 2^{\Theta}$ and the masses of x and x' in 2^{Θ} are $m_{\Theta}(x)$ and $m_{\Theta}(x')$ respectively.

In this same vein, the dominance of one belief set $\underline{A}$ over another $\underline{B}$ where both have the same common power set 2^{Θ}, is defined by

$$\underline{A} \succsim \underline{B} , \quad \text{if} \quad \sum_{x \in 2^{\Theta}} m_A(x)|x| \geq \sum_{x \in 2^{\Theta}} m_B(x)|x| , \qquad (5)$$

where $m_A(x)$ and $m_B(x)$ are the mass assignments of $\underline{A}$ and $\underline{B}$, respectively, in 2^{Θ}. Equality can also be defined similarly as

$$\underline{A} = \underline{B} , \quad \text{iff} \quad m_A(x) = m_B(x) \quad \text{for} \quad \forall x \in 2^{\Theta} . \qquad (6)$$

As defined, these relationships set up a partial order in $\underline{\Theta}$ between various power set mass distributions. The more diffuse the information content of a power set, the more dominant it is in the order. In the case of all the mass being assigned to only a single element of 2^{Θ}, two particularly useful belief sets: $\underline{E}$, with $m(\theta)=1$, and $\underline{N}$ with $m(\emptyset)=1$ can be defined. With these two new belief sets, it can be seen that, in an uncertainty context, this order is bounded by them, in that, $\inf(\underline{\theta})=\underline{N}$ and $\sup(\underline{\theta})=\underline{E}$. Also in this case of all mass being assigned to an individual element of 2^{θ}, the concept of set inclusion is a limiting case of this order, in that, if $x' \subseteq x$ then $x \succcurlyeq x'$.

Finally, the moment sums appearing in Eq.(5) can be normalized to unity by dividing by the cardinality of the possibility set to define the concept of a size or norm, / /, as

$$/\underline{\theta}/ = \sum_{x \in 2^{\theta}} m_{\theta}(x)\,|x|/|\theta| \quad . \tag{7}$$

The limits of this norm are seen to be: $/\underline{\theta}/_{max}=/\underline{E}/=1$ and $/\underline{\theta}/_{min}=\underline{N}=0$.

4) Intersection and Union Operations

The most important operators needed for any uncertainty algebra are those that allow information from various sources to effectively be combined. In DST, Dempster's rule of combination[4] is the only operator available for pooling uncertainty information. It has very strong intuitive appeal, in that it is based on both a probabilistic and set-theoretic approach. The proportionate distribution of mass between possibilities using mass products, which lies at the heart of this scheme, is a fundamental rule of combination in probability theory[4]. Set theory operations are used, on the other hand, to assign the resulting mass products to each member of the possibility power set. These two strong points make this rule the best choice for the first fundamental operator of OT, that is, the intersection operator $\oslash$. The definition of the intersection operator for the case $\underline{C} = \underline{A} \oslash \underline{B}$, where $\underline{A}$, $\underline{B}$, and $\underline{C}$ have power sets 2^{θ} with elements $a,b,c\epsilon 2^{\theta}$ and masses $m_A(a)$, $m_B(b)$ and $m_C(c)$ respectively, is therefore, given as

$$m_C(c) = \sum_{a \cap b = c} m_A(a) m_B(b) \quad . \tag{8}$$

where it is easily shown[4] that $0 \leq m_C(c) \leq 1$ and unit normalization of masses in $\underline{C}$ is retained.

In this scheme, the mass in any element of $\underline{C}$ (i.e. $m_C(c)$) is given by the sum of all the mass products in which the elements a of $\underline{A}$ and b of $\underline{B}$ intersect in c. This is the essence of Dempster's combination rule except for the fact that mass is allowed to fall into the null set $\emptyset$ if $a \cap b = \emptyset$. In DST these resulting masses are renormalized into all the other elements of $\underline{C}$ such that $m_C(\emptyset)=0$. The advantage of retaining the masses in $\emptyset$ that result from sets which have no intersection, is to have a measure of the amount of conflict which exists between the two belief sets being combined. Renormalization masks the fact that no common ground can be found to combine such information. In this context also, the mass in $\emptyset$ can also be used as measure of the incompleteness of the original possibility sets.

The choice of a suitable union operator to go along with the intersection rule above is a difficult one and involves many compromises in trying to develop the least restrictive algebra possible. When operator associativity, commutativity, unit normalization, and nonnegativity of masses are considered essential features of this theory, the choice becomes somewhat easier. In this light the mass combination rule most akin to that used in probability and group theory (i.e. $m_1+m_2-m_1*m_2$), must be rejected because it violates either one or the other of the latter two constraints mentioned above in a power set implementation. The final choice was, therefore, made using the MAX and MIN operations in FST as an analogy. That is, an upper and lower bound to the common ground for the pooling of information was formed with the union and intersection rules of set theory. This choice of the union operator $\circledvee$, for the case $\underset{\sim}{C} = \underset{\sim}{A} \circledvee \underset{\sim}{B}$, was then

$$m_C(c) = \sum_{a \cup b = c} m_A(a) m_B(b) \ . \qquad (9)$$

where again because mass products are used, $0 \leq m_C(c) \leq 1$ and the masses retain unit normalization in $\underset{\sim}{C}$.

Here, as in Eq.(8), mass products are used to distribute mass between the possibility sets to be combined, but the resulting mass, in this case, is assigned to the union of the subset of the elements being considered and not the intersection.

Use of the two basic operators $\circledwedge$ and $\circledvee$, together with the identity belief set $\underset{\sim}{E}$ and the null belief set $\underset{\sim}{N}$ gives rise to the following relationships for any belief set $\underset{\sim}{\theta}$:

$$\underset{\sim}{\theta} \circledwedge \underset{\sim}{E} = \underset{\sim}{\theta} \ , \qquad (10a)$$

$$\underset{\sim}{\theta} \circledvee \underset{\sim}{E} = \underset{\sim}{E} \ , \qquad (10b)$$

for $\underset{\sim}{E}$ defined previously as: $m_\theta(x)=0$ for $\forall x \neq \theta$ and $m_\theta(\theta)=1$, and

$$\underset{\sim}{\theta} \circledwedge \underset{\sim}{N} = \underset{\sim}{N} \ , \qquad (11a)$$

$$\underset{\sim}{\theta} \circledvee \underset{\sim}{N} = \underset{\sim}{\theta} \ , \qquad (11b)$$

for $\underset{\sim}{N}$ defined previously as: $m_\theta(x)=0$ for $\forall x \neq \emptyset$ and $m_\theta(\emptyset)=1$.

In summarizing this section, it should be emphasized, that both the union and intersection operators were defined to use the same product rule to combine masses so as to preserve commutativity and unit normalization. It is the use of the set rules $\cap$ and $\cup$ for final placement of these masses in the resulting power set that distinguished them. In this context, the use of $\textcircled{\vee}$ and $\textcircled{\wedge}$ produces upper and lower bounds to the common ground between the two belief sets being combined. This role is similar to that played by the MAX and MIN rules in FST. The OT rules, however, are not distributive in mixed operations and idempotency is also lost. This yields a somewhat less general structural base for future developments but one that appears to be necessary, given the normalization condition which ties this theory to its probability base.

5) Definition of Complementation

The last basic operation needed to complete OT is complementation. This concept can be defined by noting that this theory deals with the power set 2^{Θ} for mass assignments and not just the possibility set Θ. In a conventional power set context, every element of the power set has a complement which is also a member of the power set. To preserve mass normalization to unity, then, the complement of a belief set $\underline{\theta}$ is defined to simply shift mass between an element in 2^{Θ} and its complement in 2^{Θ} so that

$$m_{\tilde{\underline{\theta}}}(x) = m_{\underline{\theta}}(\tilde{x}) , \qquad \text{for} \quad \forall x, \tilde{x} \in 2^{\Theta} . \qquad (12)$$

That is, for the belief set $\underline{\theta}$, with mass assignments $m_{\underline{\theta}}(x)$ for $\forall x \in 2^{\Theta}$, the complement set representation $\tilde{\underline{\theta}}$, has mass assignments $m_{\tilde{\underline{\theta}}}(x)$, where $m_{\tilde{\underline{\theta}}}(x) = m_{\underline{\theta}}(\tilde{x})$ for $\forall x \in 2^{\Theta}$.

This definition gives results similar to those derived from FST in the limit of crisp sets, but in its most general form it, like FST, does not preserve the normal set-like rules for a complement. That is,

$$\underset{\sim}{\theta} \owedge \tilde{\underset{\sim}{\theta}} \neq \underset{\sim}{N} \ , \tag{13}$$

$$\underset{\sim}{\theta} \ovee \tilde{\underset{\sim}{\theta}} \neq \underset{\sim}{E} \ . \tag{14}$$

In practical application, however, the normal set results are closely approximated, as one would want. In addition, De Morgan's laws and involution, given as follows:

$$\widetilde{\underset{\sim}{A} \owedge \underset{\sim}{B}} = \tilde{\underset{\sim}{A}} \ovee \tilde{\underset{\sim}{B}} \ , \tag{15}$$

$$\widetilde{\underset{\sim}{A} \ovee \underset{\sim}{B}} = \tilde{\underset{\sim}{A}} \owedge \tilde{\underset{\sim}{B}} \ , \tag{16}$$

$$\tilde{\tilde{\underset{\sim}{A}}} = \underset{\sim}{A} \ , \tag{17}$$

are obeyed in all cases.

The proof of De Morgan's laws can easily be demonstrated using either the $\owedge$ or $\ovee$ operators. In the $\owedge$ case, for example, Eq.(8) can be used to define the operation $\underset{\sim}{A} \owedge \underset{\sim}{B} = \underset{\sim}{C}$ as

$$m_C(c) = \sum_{a \cap b = c} m_A(a) m_B(b) \ , \tag{18}$$

so that using Eq.(12), $\tilde{\underset{\sim}{C}} = \widetilde{\underset{\sim}{A} \owedge \underset{\sim}{B}}$ is then given by

$$m_{\tilde{C}}(c) = m_C(\tilde{c}) = \sum_{a \cap b = \tilde{c}} m_A(a) m_B(b) \ . \tag{19}$$

Noting that the sets a and b are dummy variables in this equation, we can switch to a complement notation to get

$$m_{\tilde{C}}(c) = \sum_{\tilde{a} \cap \tilde{b} = \tilde{c}} m_A(\tilde{a}) m_B(\tilde{b}) . \tag{20}$$

Applying De Morgan's rule to the summation index equation and Eq.(12) to the masses in the summation, yields

$$m_{\tilde{C}}(c) \;=\; \sum_{a \cup b = c} m_{\tilde{A}}(a)\, m_{\tilde{B}}(b) \;, \qquad (21)$$

which is identical to $\underline{\tilde{C}} = \underline{\tilde{A}} \ovee \underline{\tilde{B}}$. Thus De Morgan's law in the form $\widetilde{\underline{A} \owedge \underline{B}} = \underline{\tilde{A}} \ovee \underline{\tilde{B}}$, is proven. Similar manipulations prove Eq.(16).

In summary, a basic set of rules for union, intersection, and complement which can be used to manipulate power sets with mass assignments has been proposed. These rules allow operations to be performed that are not available in DST and should make further developments possible using FST as a model. Despite some similarities to FST, however, the actual results obtained with this approach will be different than those obtained by FST even in very simple cases, as is illustrated below.

6) Examples and Comparison to FST and BIT

To give some idea how the operators defined in the previous sections might be applied in an approximate reasoning problem, a simple example in logic will be worked out. This example was chosen primarily for its simplicity, but it does illustrate clearly some of the differences between OT, DST, BIT and FST. In particular, DST cannot be applied to this problem at all, for reasons to be explained, and FST produces quantitatively different results. It also has some potential applicability in expert systems, where a strict logical interpretation of implication rules with uncertainty might prove useful.

Starting with a simple, single element set $\theta=\{a\}$ and its power set $2^{\theta}=\{\emptyset,\theta\}$, the definition of the complement can be used to reinterpret this set in terms of the logical constants, T and F (i.e. true and false) if we let θ=T and $\emptyset$=F. In DST, we could go no further than this, since the consequent mass assignments would require $m_\theta(\emptyset)=0$ and therefore $m_\theta(\theta)=1$. Any further operations with such single element sets would leave these assignments unchanged. An approach to this logic problem in DST would, for example[4], have to start with a minimum of the two element set $\theta=\{T,F\}$. The relaxation of the normalization condition in OT, however, allows mass in $\emptyset$ resulting in workable rules for dealing with single element sets. These rules will be illustrated below using the logical interpretations of: $\owedge$ as AND, $\ovee$ as OR, $\tilde{}$ as NOT, and for the two belief sets $\underline{A}$ and $\underline{B}$, the operation $\tilde{\underline{A}}\ovee\underline{B}$ as the implication rule.

Assigning the following masses to the belief sets of $\underline{A}$ and $\underline{B}$ (noting the normalization condition in Eq.(1)):

$$\underline{A} = \{\overset{1-m_A}{F}, \overset{m_A}{T}\} \text{ and } \underline{B} = \{\overset{1-m_B}{F}, \overset{m_B}{T}\}, \qquad (22)$$

we can look at the consequences of operating on these sets with the logical constructs, AND, OR, NOT and implication. Note here, that only a single mass assignment is needed to completely characterize the belief sets of $\underline{A}$ or $\underline{B}$ because of the definition of the complement and the normalization condition. The results to follow will, therefore, only deal with the values of m(T) (i.e. the mass of the T power set element). The value of m(F) will always be 1-m(T).

A) AND - Equivalent to $\owedge$

In Boolean logic, the result C in the operation A AND B=C is represented by the following truth table:

		B	
		F	T
A	F	F	F
	T	F	T

Table 1. Boolean truth table for AND operator.

This table represents the limiting behavior of both OT, BIT and FST in this simple case as will be seen below.

For the OT case, using the intersection rule given in Eq.(8) and noting that the masses for T and F add up to unity, we find that the logical AND result, represented by $\underset{\sim}{A} \oslash \underset{\sim}{B} = \underset{\sim}{C}$, is given by

$$\underline{A} \oslash \underline{B} = \underline{C} = \{ \overset{1-m_A m_B}{F} , \overset{m_A m_B}{T} \} . \tag{23}$$

That is,

$$m_C(T) = m_A m_B . \tag{24}$$

This is the same result that would be obtained using BIT when multiplying probabilities p_A for A={T} and p_B for B={T} to get A*B=C. That is,

$$p_C = p_A p_B . \tag{25}$$

The equivalent result in FST is $\underset{\sim}{A} \cap \underset{\sim}{B} = \underset{\sim}{C}$ for the single element fuzzy sets $\underset{\sim}{A}$={T} and $\underset{\sim}{B}$={T}, with membership functions μ_A and μ_B, respectively. That is, the MIN operator in FST is used and

$$\mu_C = \mathrm{MIN}(\mu_A, \mu_B) \qquad (26)$$

Both the OT results for $m_C(T)$, which are also the BIT results for p_C, and the FST results for $\mu_C(T)$ are represented graphically in the form of continuous truth tables in Fig.(1).

As can be seen from this figure, the OT, BIT and FST results are functions representing the Boolean truth table given in Table 1. They all have precisely the same limits as the Boolean results (i.e. as μ, p or m approach 0 or 1), and have qualitatively similar behavior, although the OT and BIT results are smooth functions, in general, and the FST results are only piecewise continuous.

Also evident from this example, is the fact that in OT, with continued application of the $\bigotimes$ operator in a sequence of AND operations, the masses in the result will approach a limit in which $m_\theta(\emptyset)=1$ and $m_\theta(\theta)=0$. This result is typical of OT results after application of many $\bigotimes$ operations and reflects its probability basis. In BIT (i.e. probability theory), the repeated compounding of probabilities results in monotonically decreasing results and this is precisely what the OT results are duplicating. FST, on the other hand, will always be limited in this compounding effect by the smallest value of the membership function in the series; it acts like a set operator, as opposed to a probability operator. Also evident in this simple example is the almost identical quantitative and qualitative roles played by the mass and membership functions. This gives an indication that these two concepts can be related more rigorously.

B) OR - Equivalent to $\bigvee$

The results for the logical OR can be worked out in a similar fashion to those presented above. In Boolean logic the OR operation is represented again by a truth table, which in this case is

ORNL-DWG 85-16199

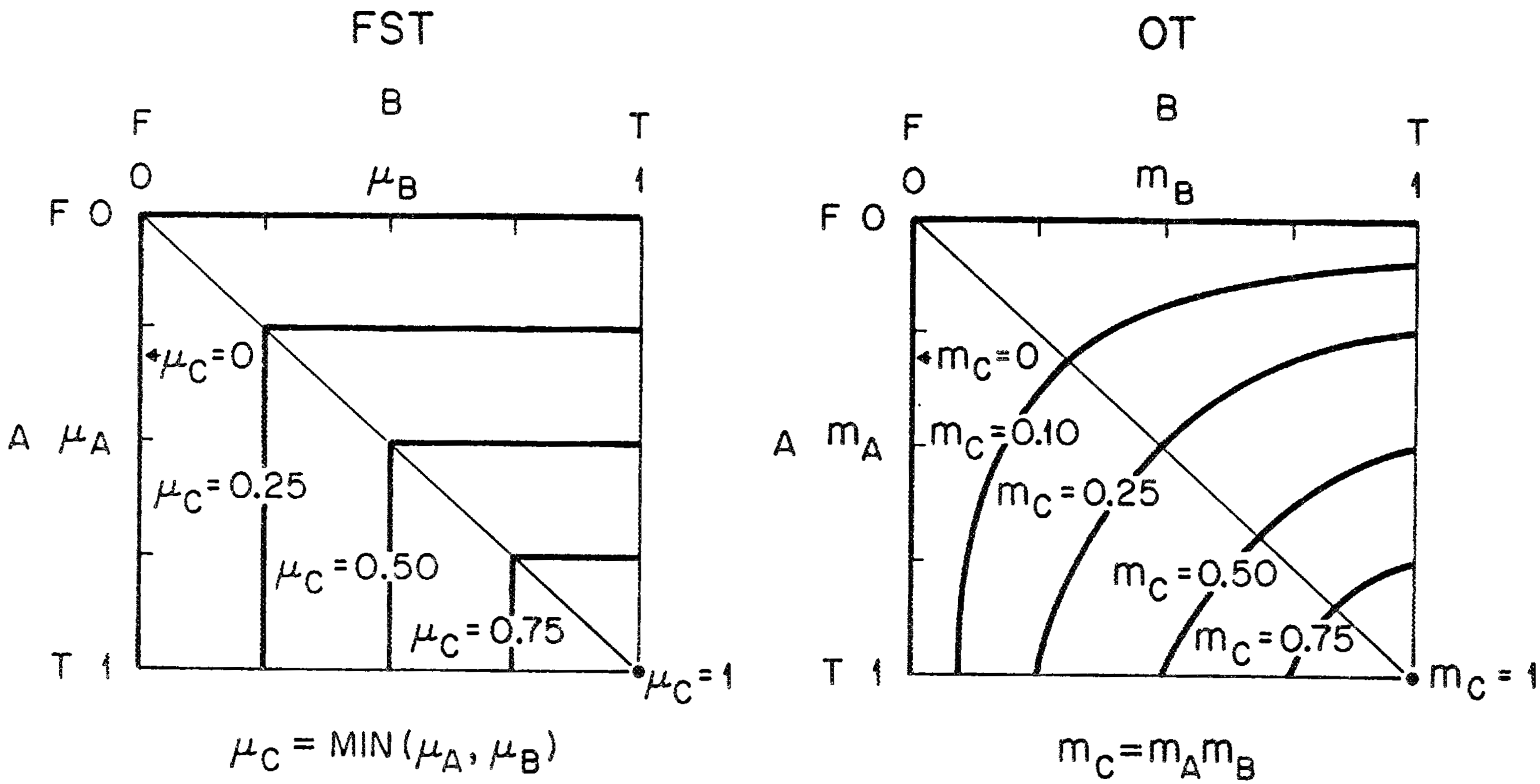

Fig. 1. Truth tables for the AND operator in OT and FST example.

		B: F	B: T
A	F	F	T
	T	T	T

Table 2. Boolean truth table for the OR operator.

In OT, using Eq.(9) for the union rule gives three mass products for $m_C(T)$. This result, rewritten using only the complement of the single mass product result for $m_C(F)$ is

$$\underline{A} \circledvee \underline{B} = \underline{C} = \{ \overset{(1-m_A)(1-m_B)}{F} , \overset{1-(1-m_A)(1-m_B)}{T} \} . \qquad (27)$$

That is, the three mass products give

$$m_C(T) = m_A m_B + m_A(1-m_B) + m_B(1-m_A) = 1-(1-m_A)(1-m_B) . \qquad (28)$$

In BIT, the equivalent of the logical OR case depends on the addition law for probabilities p_A and p_B. This law preserves normalization to unity, and for this case is A+B=C. The BIT result is then

$$p_C = p_A + p_B - p_A p_B , \qquad (29)$$

which can also be seen to be equal to

$$p_C = 1 - (1-p_A)(1-p_B) . \qquad (30)$$

The equivalence of the OT and BIT results in this case is rather interesting, in that the OT union combination rule uses only mass products to obtain its result. OT could not accommodate the use of a probability addition rule like Eq.(29) in its general formulation because of normalization constraints (see the discussion in Section 4). The same result, however, is generated by using the set union operation to store three mass products in the power set element T, which then add up to the probability rule result.

By way of comparison again, the equivalent FST results are derived from $\underset{\sim}{A} \cup \underset{\sim}{B} = \underset{\sim}{C}$, using the MAX operator. That is,

$$\mu_C = MAX(\mu_A, \mu_B) \ . \tag{31}$$

There is no need to graph the results in this case (or for that matter those for the implication rule which follow), since they are simply rotations or inversions of the same general shapes given in Fig.(1). They all have the same limiting behavior as the Boolean truth tables and the OT (and BIT) results are the smooth analogs of the piecewise continuous FST results.

Again, a typical result of OT in general, is obtained in this case with repeated application of the ⓥ operator. When this is done, the limits are seen to be the reverse of those seen in the intersection case, in that now we get $m_\theta(\emptyset)=0$ and $m_\theta(\theta)=1$. This is the same limit expected from probability theory for the addition of probabilities, as is evident from the equivalence of BIT and OT in this case. FST produces no such limit due to the nature of the MAX operation.

C) Implication - Equivalent to $\tilde{\underline{A}} ⓥ \underline{B}$

As the last part of this example, the strict logical interpretation of implication as $\tilde{\underline{A}} ⓥ \underline{B}$, will be used to highlight the three theories. Although this is a rather simplistic interpretation of implication (compared to what might be done otherwise in OT and what has already been proposed in FST[7]), it is rather instructive and computationally efficient. Current use of certainty factors in expert systems[1] is certainly on a par with this interpretation as far as computational ease is concerned, although the evidential basis of certainty factors is far more theoretically developed.

As in the last subsection, a Boolean truth table representing the limits of the continuous theories for $\tilde{A} \cup B = C$ is given by

		B: F	B: T
A	F	T	T
A	T	F	T

Table 3. Boolean truth table for the implication operator

Application of Eqs.(9) and (12) in OT to the implication rule definition gives the following results:

$$\underline{\tilde{A}} \circledvee \underline{B} = \underline{C} = \{ \overset{m_A(1-m_B)}{F} , \overset{1-m_A(1-m_B)}{T} \} . \qquad (32)$$

The value of $m_C(T)$ is a result again, in this example, of three mass products, which when simplified give

$$m_C(T) = 1 - m_A(1-m_B) . \qquad (33)$$

Noting that the complement of p_A in BIT for this case is $1-p_A$, the BIT results, using the Eq.(29) as the rule for addition of probabilities, are again seen to be equivalent to the OT results. That is, for $\tilde{A}+B=C$,

$$p_C = 1-p_A+p_B-(1-p_A)p_B = 1-p_A(1-p_B) . \qquad (34)$$

In FST, the equivalent result is obtained by taking $\tilde{A} \cup B = C$, where the complement uses a membership function of $1-\mu_A$ for T. This result is simply

$$\mu_C = \mathrm{MAX}(1-\mu_A, \mu_B) \quad . \tag{35}$$

A graphical representation of these results are again a rotation of the behavior shown in Fig.(1). All the comments made for the union case apply here as well.

Summarizing the results obtained in this simple example, it should be clear that OT and BIT are strongly connected in a probability context but differ in the way they obtain similar results. In this sense, OT uses both arithmetic and set theory operations while BIT uses arithmetic laws only. The qualitative similarities between OT and FST are also apparent, in that the former is a smooth analog of the latter.

7) General Extensions to OT

A) Connection to FST

In order to make it easier to develop additional operators and concepts in OT, it is useful to make some formal connection between the mass, m, and the membership function, μ. The examples in the last section indicate how these functions are related in a simple case and lead to the belief that a more general relationship can be found. Noting the role played by P, the upper probability in DST (defined in Eq.(3)) and the membership function μ in FST, it was felt that a formal connection could be made between these two concepts.

In DST, the function P(x) can be interpreted as the maximum possible belief in the member x of the power set 2^{Θ}. Its range for any element x is always, $0 \leqslant P \leqslant 1$, even though the masses themselves must always sum to unity. The membership function, likewise, represents a possibility (i.e. for membership) and also has the same range of values. The difference between the two, in this regard, is only that μ is defined on θ and P is defined on its power set 2^{θ}. This suggests that a formal connection can be made between μ and P by restricting P, for this discussion, to the elemental members of θ; the defining relationship is then

$$\mu(x_i) = P(x_i) = 1-B(\tilde{x}_i) = \begin{cases} 1 - \sum\limits_{x' \subset \tilde{x}_i \neq \phi} m_{\theta}(x') & x_i \neq \phi \\ 1 - m_{\theta}(\phi) & x_i = \phi \end{cases} \tag{36}$$

for $\forall x_i, \tilde{x}_i \in \theta$ and $\forall x' \in 2^{\theta}$.

Using the definition of B(x) given in Eq.(2), the same restriction (i.e. $x_i \in \theta$) can be made, giving a relationship between $\tilde{\mu}(x_i)$ and $B(\tilde{x}_i)$ which is similarly seen to be

$$\tilde{\mu}(x_i) = 1 - \mu(x_i) = B(\tilde{x}_i) \ . \tag{37}$$

Although no unique inverse relationship can be postulated between masses m(x) and membership functions $\mu(x)$ using Eq.(36), the above relationships do provide a useful way of comparing theoretical developments and results between OT (and for that matter DST) and FST.

B) Extension to mappings

As an example of the process of extending the range of applicability of OT, the definition of a general mapping rule with uncertainty will be proposed here. As with most of the developments which will be derived from this theory, an analogous path to the extension of FST will be taken. That is, set theory rules will be expanded with the role of membership functions being played by masses.

Looking first at the definition of a general set mapping rule $f: a \rightarrow b$, with $a, b \subset \Theta$. We note that, if $a_i \epsilon a$ and $b_j \epsilon b$, then the mapping f gives $b_j \epsilon f\{a_i\}$. In general then, $f\{a_i\}$ is a set with cardinality greater than unity and the mapping is characteristically an element-to-set mapping. If we now define b_a to be the union of the elements in the set $f\{a_i\}$ (the union of f being denoted by $\cup f$), we see that $b_a \epsilon 2^B$. One generalization of this mapping for OT power sets with mass assignments is now suggested (although this choice is certainly not unique).

Thus, define the general belief set mapping $F: \underline{A} \rightarrow \underline{B}$, with $\underline{A}$ and $\underline{B}$ having elements $a \epsilon 2^A$, $b \epsilon 2^B$ and masses $m_A(a)$ and $m_B(b)$, respectively, such that, for each a mapped into b, if $a \epsilon 2^A$ and $b \epsilon 2^B$, then $b \epsilon F\{a\}$. Now, letting the union of the elements of $F\{a\}$ be b_a, as before, we see that $b_a \epsilon 2^B$ and this particular function can be used to obtain the mass assignments for this mapping. That is, in assigning mass for $F: \underline{A} \rightarrow \underline{B}$, use the function $\{b_a\} = \cup F\{a\}$ so that the mass $m_B(b)$ in 2^B can be defined as

$$m_B(b) = \begin{cases} \sum_{b_a = b} m_A(a) & \forall b_a \epsilon 2^B \\ 0 & \text{otherwise} , \end{cases} \quad (38)$$

with the following mapping and function definitions:

$$F : 2^A \rightarrow 2^B, \quad \{b_a\} = \cup F\{a\} \quad \text{and} \quad a \epsilon 2^A, \; b, b_a \epsilon 2^B \; . \quad (39)$$

Since b_a is represented by the relationship in Eq.(39), it is also clear that

$$\sum_{b \epsilon 2^B} m_B(b) = \sum_{b \epsilon 2^B} \sum_{b_a = b} m_A(a) = 1 , \quad (40)$$

and the final mass distribution in 2^B is normalized to unity as it was in 2^A.

In essence, the OT mapping rule replaces the element-to-set nature of the general set mapping rule in Θ with an element-to-element function in 2^Θ. This route was taken so that a normal set mapping rule could be used directly in OT without modification of its definition to distribute set masses appropriately. The rule-of-thumb used here was: collect mass into a set common to all the elements mapped into if no further information was available from the mapping definition to do otherwise. This decision converts mappings to functions, preserves normalization to unity and gives rise to a larger class of mappings in OT which will have unique inverses. The generalized rule should also provide another means of attacking the problem of uncertainty propagation in expert systems, in that the implication rule can alternatively be though of as a general mapping (i.e element-to-set). Future development in OT will be required, however, to bear this conjecture out.

Before concluding, a simple example of an OT mapping will be given to make the concept easier to understand in practice. For the case of $\Theta=\{x_1,x_2\}$, the particular mapping example: $F:\underline{A}\rightarrow\underline{B}$ with $\underline{A}$ and $\underline{B}$ having elements in 2^Θ and respective mass assignments $m_A(x)$ and $m_B(x)$, will be represented in 2^Θ by the following figure:

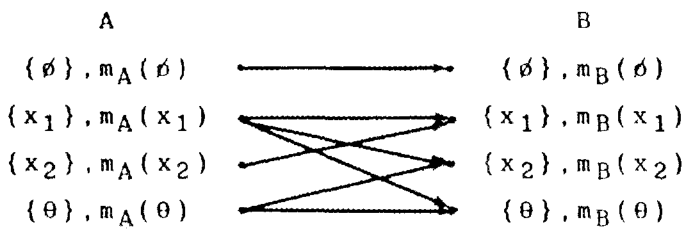

Figure 2. OT representation of the mapping $F:2^A\rightarrow2^B$.

and the element mapping rule

$$F\{\phi\}=\{\phi\},\quad F\{x_1\}=\{x_1,x_2,\theta\},\quad F\{x_2\}=\{x_1\},\quad F\{\theta\}=\{x_2,\theta\}\quad . \qquad (41)$$

Using the OT mapping rule given in Eqs.(38) and (39), this mapping gives rise to the particular function b_a, in which the following relationships hold:

$$b_\phi=\{\phi\}=\cup F\{\phi\},\quad b_1=\{\theta\}=\cup F\{x_1\}\quad ,$$
$$b_2=\{x_1\}=\cup F\{x_2\},\quad b_\theta\{\theta\}=\cup F\{\theta\}\quad . \qquad (42)$$

If b_a is used now to distribute masses, the following figure can be used to represent this function and its final results:

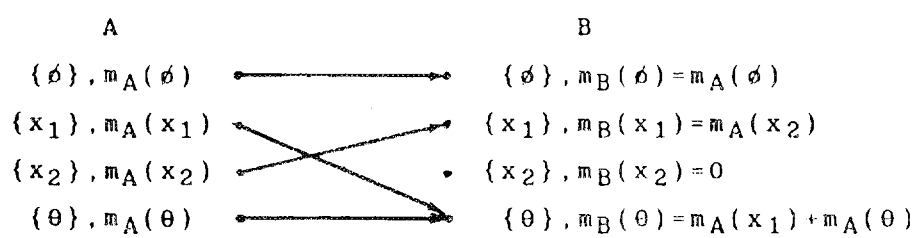

Figure 3. Final OT results for the mapping $F:2^A\rightarrow 2^B$.

Note here that the final form of the mapping is a simple function and the mass distribution in 2^B is normalized to unity. Also, as alluded to previously, this simple case might be useful in expert systems if the set θ were chosen to be $\theta=\{T,F\}$ and the implication rule translated using such a general mapping.

8) Conclusions

A hybrid uncertainty theory has been developed to bridge the gap between fuzzy set theory and Bayesian inference theory. Its basis is the Dempster-Shafer formalism (a probability-like, set-theoretic approach), which has been extended and expanded upon so as to include additional basic operations for manipulating uncertainties in approximate reasoning. The new theory, operator-belief theory (OT), retains the probabilistic flavor of Bayesian inference but includes the potential for defining a wider range of operators like those found in fuzzy set theory.

The basic operations defined for OT in this paper include those for: dominance and order, union, intersection, complement and general mappings. A formal relationship between the membership function in fuzzy set theory and the upper probability function in the Dempster-Shafer formalism was also developed. Several sample problems in logical inference were worked out to illustrate the results derived from this new approach as well as to compare them with the other theories currently being used. A general method of extending the theory using the historical development of fuzzy set theory as an example was suggested.

Future development of OT will concentrate on devising efficient computational algorithms for its implementation in expert or rule-based system applications. The OT union and intersection rules seem to have a natural basis in matrix algebra and are highly suitable for implementation in concurrent algorithmic form on a hypercube computer. Additional work is also needed in defining suitable projection operators for making decisions on the basis of power set mass assignment results. Definitions for suitable direct addition and subtraction operators are also needed so that evidence and belief can be gathered and combined to form an initial power set mass assignment.

The theory will be extensively tested in its current form as part of the Oak Ridge National Laboratory CESAR program in robotics and machine intelligence[9]. It has applicability in this program's planning, sensor fusion, vision and expert system efforts.

REFERENCES

[1] B. G. Buchanan and E. H. Shortliffe, Rule-Based Expert Systems (Addison-Wesley, California, 1984), sects-11,12.

[2] G. Polya, Mathematics of Plausible Reasoning, Vol II: Patterns of Plausible Inference, 2^{nd} ed. (Princeton University Press, N. J., 1968).

[3] A. Kaufmann, Introduction to the Theory of Fuzzy Subsets, Vol 1 (Academic Press, N. Y., 1975).

[4] G. Shafer, A Mathematical Theory of Evidence (Princeton University Press, N. J., 1976).

[5] J. D. Lowrance and T. D. Garvey, Evidential reasoning: A developing concept, in Proc. IEEE Int. Conf. Cybern. Soc., 1982.

[6] ibid [1], chapter 13.

[7] H. Prade, A computational approach to approximate and plausible reasoning with applications to expert systems, IEEE Trans. Pattern Anal. Mach. Intel., Vol. PAMI-7, May 1983.

[8] C. R. Weisbin, et al., Machine intelligence for robotics applications, in Proc. Conf. Intel. Sys. & Mach., Oakland University, Minn., April 22-24,1985.

[9] CESAR - Center for Engineering Systems Advanced Research, Oak Ridge National Laboratory, Oak Ridge, Tenn. (see [8] for details of program).

SUMMARIES

SUMMARIES

DOE/CESAR WORKSHOP: "PLANNING AND SENSING FOR AUTONOMOUS NAVIGATION"

August 18-19, 1985

Jacob Barhen
Oak Ridge National Laboratory

Question: What are the most suitable computer architectures for the "brain" of an intelligent machine?

Two major points emerged from the discussion:

1. No single architecture can be expected to optimally satisfy the conflicting reqirements of the various computational activities which need to be carried out onboard the mobile intelligent robot. Three possible architectural approaches were suggested:

   - homogeneous ensembles, including
     - coarse grain machines (ORNL hypercube, butterfly machines),
     - fine grain machine (connection, rex machines),
     - very coarse grain,
   - inhomogeneous machines, including
     - "star" configuration (central master) (Yutaka Kanayama),
     - fully distributed (Scott Harmon) and
   - special purpose "chips." (Crowley).

Current experience lies mainly with inhomogeneous configurations; such systems built from existing "off-the-shelf" technology appear to handle rather well the tasks assigned to them so far. However, the complexity of these tasks is rather limited compared to ultimate requirements for a machine to operate autonomously in unstructured environment. The consensus is that much "hands-on" experience is needed with the various architectures before a firm commitment can be made.

2. Very high power concurrent computers (at low costs) are no longer in the realm of science fiction. As pointed out by Alex Parodi, we need to adopt a new mental attitude, i.e., "think concurrent."

Summary: VLSI is coming of age; VLSI-based concurrent computation is the only known way to obtain vast increases in computing power at reduced costs.

New methods need to be developed to deal with robotic problems in real-time on such concurrent systems. These include the development of an adequate system's environment (operating system) to provide a generalized framework for implementing machine intelligence components. Some initial ideas were proposed to address the class of problems dealing with the co-ordination of synchronous distributed processes. These ideas include process management, message communication and synchronous dynamic load balancing and virtual time methodology.

DOE/CESAR WORKSHOP: PLANNING AND SENSING FOR AUTONOMOUS NAVIGATION"

August 18-19, 1985

Professor James Crowley
Carnegie Mellon University

Question: How can world mapping and discovery best be accomplished in unstructured environments?

The question was interpreted as a call for a taxonomy of techniques for world mapping in an unstructured environment. The taxonomy was organized according to the way in which the system would process data from sensors. To focus the discussion the group was given an agenda with time limits.

- Definitions (10 minutes)
- List Taxonomy (20 minutes)
- Discuss Techniques (30 minutes)

The following is a summary of the discussion and conclusions for each agenda item.

1. Definitions

Ten minutes were allocated to consider what the terms of the question meant and what the question called for.

The crucial terms in the question were seen as "world mapping," "discovery" and "unstructured environment." Two terms relating to procedure were "how" and "best be accomplished." How was seen as a call for a taxonomy of techniques. It was proposed to discard the term best, as best can only have meaning with regard to a specific set of constraints.

Unstructured environment was at first seen to have two meanings:

1) no prior knowledge exists about the environment, and

2) the elements in the environment are constantly changing. It was observed that the laws of physics provide a basic structure for environment. Static vs dynamic elements were observed to be degrees of simplicity. That is, the environment might include objects in motion. If so, the motion may need to be explicitly modeled.

World mapping was (after some discussion) defined to be a process of building a description which may include

- places and their connectivity,
- geometric primitives, and
- objects.

Places, geometric elements and objects have attributes of location, and may include other attributes.

Discovery was seen to be a snyonym of world mapping.

2. Taxonomy

The discussions brought out a taxonomy arrayed according to what the system did with raw sensor data. The taxonomy included:

- Followers,
- Geometric Modelers,
- Objects Discoverers, and
- Strategy Learners.

Followers use raw sensor data to "follow" structures such as walls, roads, buried wires, etc. The sensor data is immediately discarded. The result produced by a follower would be a network of places that can be traveled by the robot. Followers may follow static objects (walls) or moving objects (people, cars, etc.)

Geometric Modelers construct a description of the environment in terms of geometric primitives. These can include

- 2D limits to free space,
- 3D surface patches, and
- Primitives such as cylinders, cubes, and surfaces.

More abstract primitives can permit inferences about function. A Geometric Modeler may use the description to construct a map of the decision points for path planning. Path planning may also be accomplished by "following" or acting directly on the geometric model.

Object Discoverers collect ensembles of geometric primitives into entities called "objects." Objects can then be "generalized" into classes. Properties can be learned by experience with object classes. Hierarchies of objects and sub-objects can be constructed.

Strategy Learners learn methods for acting on objects. Examples may include "a box may be picked up by its handle" or "a tank may cross the river at this place" or "do not ride a bicycle on this road."

Conclusion: The sophistication of world modeling needed for a mobile robot depends on the goals which must be accomplished. A mail delivery robot can get by nicely by following a buried wire. A maintenance robot must know about objects, their parts, and how they interact to accomplish its tasks.

DOE/CESAR WORKSHOP: "PLANNING AND SENSING FOR AUTONOMOUS NAVIGATION"

August 18-19, 1985

Stan Rosenschein
Leslie Kaelbling

Summary of Discussion on questions 2 and 4

In this session we discussed several promising approaches for real-time planning with sensor feedback from execution. There was general agreement on several major points: "canned," straight-line programs are of little interest in mobile robotics; many interesting problems cannot today be solved in real time; high-level and low-level planning processes may proceed at different time scales, but must be coordinated if timely response to environmental events is to be guaranteed.

Detailed discussion centered around topics which cluster into two broad categories: getting information from the environment and using information to guide actions. The following is a summary of problems in each area, along with several proposed solutions.

Getting Information

1. (Cost of sensors) It is not always economical to equip mobile robots with the most suitable sensors.

    - This problem should diminish with time and with mass production arising out of increased utility.

2. (Limited sensors) Sensors typically return weak information, often in large quantities.

    - For simple tasks, limited information is often sufficient.
    - Multiple sensor readings can be integrated to provide more precise information.
    - For some tasks, specialized sensors are technically and economically feasible.

3. (Poor models) It is difficult to model complex physical systems, such as sensor systems and task environments.

    - As a first step, formal methods should be adopted for representing information states. Two candidate formalisms are Bayesian (probabilistic) models and standard logic and model theory.

- Some opportunities may exist for automatically improving models through learning.

4. (Poor interpretation techniques) Even with better sensors and models, problems of sensor integration will remain due to poor techniques for analyzing and interpreting the data.

   - Much recent progress in low-level perception has resulted from careful analysis of constraints inherent in physical systems. This analysis suggested improved algorithmic techniques. This approach should be refined and applied to a broader range of interpretation problems.

Using Information

1. (Inadequate specification languages) Specifications of desired robot behavior are typically given incompletely and informally.

   - Structured, but informal specification languages currently exist. Formal methods for describing robot behavior may be based on these, or adapted from techniques developed in other branches of computer science.

2. (Relating information to action) It is difficult to map complex representations of a situation to actions which are appropriate (relative to an objective) to that situation.

   - For specialized classes of situations and goals (e.g., navigation by map) particular techniques exist for transforming situational information into actions (e.g., graph search).

   - For more complex transformations, hierarchical methods are required.

   - An important class of actions are those performed with the intent of acquiring new information.

August 22, 1985

DOE/CESAR Workshop: "Planning and Sensing for Autonomous Navigation"
August 18-19, 1985, Los Angeles, CA

Dr. Peter Cheeseman - NASA/AMES

QUESTION: "What are the most promising approaches toward uncertainty representation and propagation?"

Answer: Bayesian methods augumented with Maximum Entropy (i.e. probabilistic methods) are the optimal representation (procedure) for making decisions under uncertainty. There are certain decision making situations (where it is possible to obtain more information) where probabilities are necessary - that is, a measure of the uncertainty of the probability value. Such second order probabilities bear a resemblance to Dempster-Shafer methods, but the rules of combination are different and lead to different results. Also, Dempster-Shafer theory does not provide an interpretive framework (i.e., how to map data into intervals and how to use the intervals for decision making. Second-order probability methods are the best (optimal) representation (procedure) in situations where they are appropriate (unless it is necessary to go to higher order probabilities).

NOTES:

(1) Part of the difficulty in answering this question is uncertainty about what? Since this is a robot workshop, one of the most important items of information about which the robot is uncertain is locational uncertainty. This includes position (e.g., x, y, z) and orientation. Provided that the errors associated with the robot movements are small and that locational information is combined in a network of relative transformations (with uncertainty) there is a first order approximation solution. This solution uses covariance matrices to represent the locational uncertainty, and a complete calculus for computing the uncertain relationships between any two objects is described in Smith and Cheeseman (1985).

(2) Another form of uncertainty concerns constructing the most probable model of the world using the available prior knowledge and the (noisy) data obtainable by the robot. Again the solution to this problem is a probabilistic/Bayesian method.

The basic idea is as always to combine whatever prior knowledge is available (e.g., detailed maps or weak information, such as "planar surfaces") with information from (noisy) sensors. The results of such a Bayesian analysis may be a very weak model with a few surfaces reasonably certain and insufficient information for other regions. That is, it only provides as strong a model as the data suggests. The main advantage of Bayesian methods is that they have a built in bias against complex (i.e., a priori unlikely) models, and only accepts them if the evidence really supports such a possibility.

Some of the difficulties in applying Bayesian methods for model discovery include the difficulty in defining what the problem is. Many of the (spurious) arguments used against Bayesian (probabilistic) approaches amount to an apparent ambiguity in specifying the prior probabilities, however, such situations are not an argument against Bayesian, but only show that the problem is still undefined. Another difficulty relates to the problem experts have in estimating their prior probabilities, but this should not be a problem for the robot, since its memory (past experience) is explicit.

(3) The use of Bayesian (probabilistic) methods is controversial, and many other methods of representation of uncertainty have been proposed (e.g., Dempster/Shafer, Fuzzy Sets. However, these methods lack an explicit interpretive framework which shows how to map data into the representation making behavior. Without such a framework, these alternative methods will continue to generate unsupportable claims and arguments that cannot be resolved. Until the proponents of alternative uncertainty formalisms provide an explicit interpretive framework and show that the result is superior to the well known and tested Bayesian methods they should cease to make unsupported claims. (See attached "Challenge.")

CHEESEMAN'S CHALLENGE

CHEESEMAN'S CHALLENGE

August 22, 1985

CHALLENGE

"To those who believe that Bayesian methods (augmented with Maximum Entropy) do not lead to optimal decision making behavior under uncertainty, please show at least one example where non-optimal behavior occurs. To prevent ambiguity concerning subjective probabilities, the example must concern a robot whose memory contents (prior experience) is known explicitly, and the robot must make an explicit decision with incomplete knowledge and limited sensor capability (i.e. a feasible robot, even if somewhat idealized), to achieve given goals."

NOTES

(1) "Not an optimal decision" means that there exists another procedure (to be given) which uses the same information which can be shown on average to satisfy the given goals more often. This definition has the following consequences:

- It avoid debates about whether any intuitions about optimal behavior is better or worse by making the criterion the average goal satisfaction, where goals are preferred.
- The on average requirement is to avoid a strong dependence on the particulars of the chosen example.
- The same information requirement is to avoid spurious arguments about the robot's behavior being non-optimal because "if it knew A, then it should not have done what it did." (Hindsight arguments.)
- The example should make clear how sensor information is to be used and represented.

(2) If no such example can be found, then the conclusion is that Bayesian methods are the optimal procedure for decision making under uncertainty (at least empirically).

Anyone who first meets this challenge will receive $50.00 (or lose $50.00 if the example fails the test). Differences of opinion will be resolved by CESAR management.

August 27, 1985

C. R. Weisbin

Peter Cheeseman's Challenge

It is impossible to answer Cheeseman's challenge because he does not define what he means by "Bayesian methods (augmented with Maximum Entropy)" nor by "optimal decision making behavior under uncertainty."

It is well known that some interpretations of Bayesian non-informative prior may lead to contradiction, as illustrated by the example below.

I don't know how to make explicit the memory contents of a robot. But it is possible to imagine a feasible robot, even if somewhat idealized, with the same memory content as R. Von Mises when it proposed its wine/water paradox. If faced with the task of determining the most probable speed of an object, as described in the example below, such a robot may be perplexed by the ambiguity and may be lead (as were Von Mises and several of his statistician colleagues), to abandon Bayesian methods altogether.* I am not sure that this would qualify as "non-optimal behavior."

Several authors have attempted to deal with the Von Mises paradox. The well known "Desideratum of Consistency" of Jaynes (Probability Theory, 1974) is somewhat "ad hoc." In my opinion, the only resolution of the paradox has been proposed by F. G. Perey (ORNL-5908, 1982). Whether Perey's theory qualifies as a Bayesian Method (augmented with Maximum Enthropy) or is a new development is for CESAR management to resolve (see note below).

Note that in F. G. Perey's theory only $\bar{v}_1$ qualifies as a most probable velocity, because the quantity directly observed is an instant of time for which the volume measure in the group manifold is constant (see F. G. Perey p. 27).

Gerard de Saussure

GdS:sar

---

* If abandoning Bayesian methods qualifies as nonoptimal behavior, I have given an example where Bayesian methods lead to nonoptimal behavior and hence met the letter (if not the spirit) of the challenge. On the other hand if abandoning Bayesian methods qualifies as optimal behavior I have certainly met the spirit (if not the letter) of the challenge. In either case, remember that I agree to return $25 to CESAR management.

Noninformative Prior Paradox

Assume we want to determine the speed of an object. We know the velocity of that object to be constant, but don't know its value. So we could place two marks, separated by a distance d, along the trajectory of the object, start a digital clock when the object passes the first mark and stop the clock when it passes the second mark. We now know that the object passed the first mark at t=0 and the second at $t_1 \leq t \leq t_2$. Since the clock is digital there is a small uncertainty, one clock oscillation, about the time at which the object reached the second mark.

We now use Bayes Theory to determine the most probable speed of the object.

First Approach

Determine the most probable time, $\bar{t}$, at which the object passes the second mark. We use a noninformative (hence constant) probability distribution for t:

$$p(t)dt = \begin{cases} dt/(t_2 - t_1) & \text{for } t_1 \leq t \leq t_2 \\ 0 & \text{otherwise} \end{cases}$$

$$\bar{t} = \int_{t_1}^{t_2} t\ p(t)dt = \frac{1}{2}(t_1 + t_2)$$

hence the most probable speed is:

$$\bar{v}_1 = \frac{d}{\bar{t}} = \frac{2d}{t_1 + t_2}$$

Second Approach

We know that the speed v is $d/t_2 \leq v \leq d/t_1$ we compute the most probable value $\bar{v}_2$ by now using a non-informative prior on v:

$$p(v)dv = \begin{cases} \dfrac{dv}{d/t_1 - d/t_2} & \text{for } \dfrac{d}{t_2} \leq v \leq \dfrac{d}{t_1} \\ 0 & \text{otherwise} \end{cases}$$

$$\bar{v}_2 = \int_{d/t_2}^{d/t_1} vp(v)dv = \frac{d}{2}\left[\frac{1}{t_1} + \frac{1}{t_2}\right].$$

In general, $\bar{v}_1 \neq \bar{v}_2$. This is because in the first approach the noninformative prior is taken over t, whereas in the second approach it is taken over v (or 1/t). Bayesian methods, by themselves, do not specify how to choose between the two approaches. (The above example is a transposition of the well known wine/water paradox of Von Mises, Probability, Statistics, and Truth, 1928.)

October 1, 1985 (revised October 15, 1985)

C. R. Weisbin

<u>Peter Cheeseman's Challenge</u>

It is impossible to answer Cheeseman's challenge because he does not define what he means by "Bayesian methods (augmented with Maximum Entropy)" nor by "optimal decision making behavior under uncertainty."

It is well known that some interpretations of Bayesian non-informative prior may lead to contradiction, as illustrated by the example below.

I don't know how to make explicit the memory contents of a robot. But it is possible to imagine a feasible robot, even if somewhat idealized, with the same memory content as R. Von Mises when it proposed its wine/water paradox. If faced with the task of determining the best estimate of the speed of an object, as described in the example below, such a robot may be perplexed by the ambiguity and may be lead (as were Von Mises and several of his statistician colleagues), to abandon Bayesian methods altogether.* I am not sure that this would qualify as "non-optimal behavior."

Several authors have attempted to deal with the Von Mises paradox. The well known "Desideratum of Consistency" of Jaynes (Probability Theory, 1974) is somewhat "ad hoc." In my opinion, the only resolution of the paradox has been proposed by F. G. Perey (ORNL-5908, 1982). Whether Perey's theory qualifies as a Bayesian Method (augmented with Maximum Enthropy) or is a new development is for CESAR management to resolve (see note below).

Gerard de Saussure Gérard

GdS:sar

---

* If abandoning Bayesian methods qualifies as nonoptimal behavior, I have given an example where Bayesian methods lead to nonoptimal behavior and hence met the letter (if not the spirit) of the challenge. On the other hand if abandoning Bayesian methods qualifies as optimal behavior I have certainly met the spirit (if not the letter) of the challenge. In either case, remember that I agree to return $25 to CESAR management.

Hi Chuck:

This note is a reply to the G. deSaussure who responded to my challenge to produce an example of non-optimal Bayesian behavior. What he produced is the same tired example with slightly different window dressing. This example surfaces over and over again, in fact his reply starts off, "It is well known that some interpretations of Bayesian non-informative prior may lead to contradiction...", which is NOT true. What IS true is that those who have not taken the trouble to read the recent literature (see enclosed paper) and just repeat the arguments of philosophers (and others) believe that Bayesian methods have been discredited, even though none of the proported examples hold up under close examination. The reason for the challenge is for those that have these beliefs to "put up or shut up." Fortunately, Gerard has provided a particular example making it possible to show exactly where his error is. It is those that rely on "it is well known..." type arguments that are impossible to deal with.

Let me first respond to the specific example, then to the more general comments in Gerard's reply.

The solution to the problem of providing the best probability estimate (i.e., that which assumes the least information--not "the most probable") of the velocity of an object based on a digital clock is exactly as described in Gerard's "First Approach." That is, the information that the clock provides is that the event "the object passed the second mark" occurred at some time t between t1 and t2. Since what was MEASURED was the relevant time interval, we have no information about when the event occurred within this interval, and so the only unbiased estimate of the probability density within this interval is the uniform distribution. The mean of this distribution is:

$$t(mean) = 1/2 * (t1 + t2)$$

and so the most unbiased estimate of the velocity IS:

$$v(estimate) = 2d/(t1 + t2).$$

At this point many say why not put a uniform distribution on log(t) or 1/t or sin(t) or any other function the user is interested in? The answer to this apparent arbitrariness is that as long as the parameter is an undefined entity, then there is no reason for preferring one function over another. However, in this case, the parameter is time which we know to be a linear single valued parameter, and we have an instrument (a clock) which directly measures this parameter. Thus, we no longer have the freedom to pretend that the clock is measuring some function of time, which is exactly what Gerard is doing in his Second Approach. Because he is interested in the velocity, he assumed a uniform distribution over the possible velocity range EVEN THOUGH THE CLOCK IS NOT MEASURING A VELOCITY. Assuming a uniform velocity distribution over the time interval is equivalent to assuming a uniform distribution over d/t, since v = d/t. Since this is not what the clock is measuring, then the second approach is just WRONG---not a contradiction as claimed.

If, instead of a clock, a Doppler shift velocity meter was used that records which frequency range the observed reflected signal belonged to, Gerard's second approach would be correct and the other incorrect. This is because a frequency range is directly proportional to velocity, and since the instrument is measuring a frequency range, the only unbiased estimate of the frequency (if a single number must be given) is the mid-value of the range. If the instrument operated on say a logarithmic scale, then a uniform distribution or log(f) would be correct instead.

As long as the definition of a problem is vague, ambiguous, or incomplete, then it is impossible to assign prior probabilities, as indicated by Bertrand's "paradox." This is not a weakness of the theory as many have maintained, but is similar to the situation in Physics where a particular set of differential equations don't have a solution until the boundary conditions are specified. No one advocates that we throw out physics because we cannot solve incompletely specified problems, but that is exactly what those who reject Bayesian methods are doing. Once the various parameters of a system are identified with measurable quantities, and the instruments for measuring these quantities have been chosen, all the degrees of freedom are (usually) used up and the problem becomes determinate.

If the prior information is weak, then the probability distribution deduced from it will also be weak (i.e., relatively flat). The accuracy of the probability (the mean ofthe distribution) once all the information has been digested is indicated by how peaked the associated probability distribution is around the mean value. Consequently, an interval or other representation designed to give a measure of the accuracy of the probability is unnecessary.

<u>NOTES</u>

1. I am not familiar with F. G. Perey's work, but since he gets the same answer (the first approach is the only correct one for the particular problem), then he is probably also on the right path.

2. How anyone can say Jayne's "desideratum of consistency" is "ad hoc" is quite beyond me! Consistency is consistency, and if you don't like any particular desideratum, then

propose an alternative. The desideratum are fully explicit and sufficient to derive all of probability theory--how could such fundamental requirements be ad hoc????

3. On the question of whether the use of logic to find the appropriate priors (including the use of group invariants methods as in the attached paper) qualifies as "Bayesian methods" is a question of definition. Since Bayesian methods require priors before they can be applied, I assumed that techniques for deciding the priors would be included under the heading "Bayesian Methods" even though I didn't say so in the original (brief) challenge. Sorry for any confusion this lack of clarification might have caused.

Well Chuck, it is up to you....a challenge was issued and accepted. This reply is the answer to that acceptance, and your decision on whether claims in the challenge still stand or have been refuted is eagerly awaited. Please feel free to ask for further clarification if you think it will help.

Peter

P.S. If you decide that I have won, please donate the money to a charity of your choice.

october 7, 1985

From: F. G. PEREY

To: C. R. WEISBIN

Subject: CHEESEMAN's CHALLENGE.

Via his CHALLENGE, Peter Cheeseman makes in essence two broad claims for what he calls "Bayesian methods (augmented with Maximum Entropy)": 1) they are free of contradictions and, 2) they are complete in the sense of being able to deal with any situations a robot could face.

One major difficulty with Peter Cheeseman's CHALLENGE is that what are "Bayesian methods (augmented with Maximum Entropy)" is not specified. Can one presume that what Peter Cheeseman means by this are E.T. Jaynes' Bayesian methods ? On the assumption that this was what Peter Cheeseman meant, I was ready to pick up the CHALLENGE because Jaynes methods are incomplete in the sense that there are certain situations in which they cannot be used and Peter Cheeseman may have been at a loss to propose a Bayesian solution that would be free of contradictions. Before I had a chance to challenge Peter Cheeseman with one of these situations, Gerard Desaussure had picked up the CHALLENGE and presented a situation where he stated that a particular solution was the correct Bayesian solution according to me. I became aware of this after Gerard had sent his letter, and I pointed out to him that he was wrong. I fully expected Peter Cheeseman to crucify Gerard with the correct answer to this problem, but to my amazement he did not do so and fully agreed with the solution that Gerard had wrongly said was the correct one. Furthermore, in his answer to Gerard, Peter Cheeseman gave some details of what he considered to be the correct Bayesian solution to a broad class of problems. We, therefore have no longer to assume what Peter Cheeseman means in his CHALLENGE by Bayesian methods since he has explicitly stated in his answer to Gerard what they are in a broad class of problems.

It is the purpose of this note to point out that what Peter Cheeseman advocates is a correct Bayesian procedure leads to contradictions. In view of the fact that in his answer to Gerard, Peter Cheeseman objects strongly to Gerard finding Jaynes Desideratum of Consistency ad hoc, I will presume that Peter Cheeseman will not object if I use it, and consequently Jaynes' terminology, to show that there is a contradiction in what he, Peter Cheeseman, advocates should be done by the robot.

Let me rephrase Gerard's problem in Jaynesian terminology. The robot knows that the distance between two markers on a rail is precisely x, that is to say this information is in his memory. The robot also knows that on this rail some objects move only at a constant velocity and this robot must make a decision depending upon the velocity of an object moving on this rail. For this purpose the robot has been equipped with sensors so that he can determine the velocity of objects moving on this rail. We will assume that physics, mathematics and decision theory are also known to the robot. Consequently the robot will have a loss function associated with any decision he must make. In order to minimize his expected loss in deciding what is the velocity of the object moving on the rail, the robot will have to assign a density function to the velocity of the object he determines from his sensors. What Peter Cheeseman states unequivocally in his answer to Gerard is that if the robot sensor is a clock that determines the time interval, t, for the object to travel the distance x to be: $t1 < t < t2$; then the robot must assign a uniform probability density function:

$$P(t)dt=dt/(t2-t1), \text{ for } t1 < t < t2,$$

to the time the object took to cover the distance x.

According to Peter Cheeseman the robot should then decide upon the mean of this distribution as the time the object took:

$$t(mean) = (t1+t2)/2$$

and, Peter Cheeseman goes on, so the most unbiased estimate of the velocity is:

$$v(estimate) = 2d/(t1+t2).$$

Let me at this stage quote Peter Cheeseman: " At this point many say why not put a uniform distribution on log(t) or 1/t or sin(t) or any other function the user is interested in? The answer to this apparent arbitrariness is that as long as the parameter is an undefined entity, then there is no reason for preferring one function over another. However, in this case, the parameter is time which we know to be a linear single valued parameter, and we have an instrument (a clock) which directly measures this parameter. Thus, we no longer have the freedom to pretend that the clock is measuring some function of time, which is exactly what Gerard is doing in his Second Approach. Because he is interested in the velocity, he assumed a uniform distribution over the possible velocity range, THOUGH THE CLOCK IS NOT MEASURING A VELOCITY. Assuming a uniform velocity distribution over the time interval is equivalent to assuming a uniform distribution over d/t, since v=d/t. Since this is not what the clock was measuring, then the second approach is just WRONG---not a contradiction as claimed.

If, instead of a clock, a Doppler shift velocity meter was used that records which frequency range the observed reflected signal belonged to, Gerard's second approach would be correct and the other incorrect. This is because a frequency range is directly proportional to velocity, and since the instrument is measuring a frequency range, the only unbiased estimate of the frequency (if a single number must be given) is the mid-value of the range. If the instrument operated on say a logarithmic scale, then a uniform distribution on log(f) would be correct instead."

I will now show that what Peter Cheeseman says should be done leads to a contradiction. Peter Cheeseman agrees with the fact that v=x/t, I use x instead of d to avoid confusion about differentials such as dv and dt. If he takes the probability for t to be in the interval dt about t to be:

P(t)dt=dt/(t2-t1), for t1 < t < t2, ...(1)

(sometimes referred to as the Laplace distribution); then he must have no objection if I use the transformation v=x/t to obtain a probability distribution for v to be in the interval dv about v. Doing the trivial algebra one obtains:

P(v)dv=dv/((v1-v2)*v**2) , for v2 < v < v1, ...(2)

where v1=x/t1 and v2=x/t2.

From a purely mathematical point of view, given that v=x/t and x is a constant in this problem because it is taken to be perfectly well known, the probability distributions (1) and (2) are identical. When Peter Cheeseman advocates taking the mean of t over (1) as the time it took for the object to cover the distance, why does he not advocate taking the mean of v over (2) for the velocity of the object? Why does he states that the correct answer for v based upon (1) is: 2d/(t2+t1)?

Assuming that Peter Cheeseman subscribes to Jaynes ideas on decision theory, let me make here a disgression. One makes the decision to choose the mean of a distribution when one has a loss function which is said to be linear in the difference between the true value and the value chosen, i.e. it is proportional to |v(chosen)-v(true)|. However, one chooses the expectation value of the distribution when one's loss function is said to be quadratic, i.e. it is proportional to (v(chosen)-v(true))**2. I do not care about which loss function the robot has in this problem for choosing a particular value of v given a probability distribution for v, he must stick to whatever it is , and if it is linear he will choose the mean, but if it is quadratic he will choose the expectation value. Because it is mathematically simpler to obtain it I will assume that the robot will choose the expectation value of the distribution, i.e. Jaynes would say his loss function is assumed to be quadratic.

What is the mistake that Peter Cheeseman has made in going from:

t(mean)=(t2+t1)/2,

to

v(estimate)=2d/(t1+t2),

assuming he wanted to take the expectation values? Denoting by <v> the expectation value of the probability distibution of v, what he has done is simply equated <v> with x/<t>. Now, given that v=x/t, where x is a constant, the mathematically correct result, and this has nothing to do with Bayesian methods, is that: <v>=x*<1/t>.

I have not yet shown that what Peter Cheeseman advocates leads to a contradiction. I have merely pointed out a trivial mathematical mistake he has made. What I have precisely shown so far is that if Peter Cheeseman says the correct Bayesian thing to do is to take the probability distribution (1); then he must grant me that it leads to the distribution (2) for the velocity. If Peter Cheeseman says that given (1) one must choose for the time t the mean value of (1), or its expectation value; then he must agree that for the velocity one must choose the mean value of (2), or its expectation value. Where Peter Cheeseman gets into trouble is in denying that having measured with a 'clock' that the time t was: t1 < t < t2, it is correct to say that we have then determined that v2 < v < v1, where v2=x/t2, v1=x/t1, and v=x/t. The reason he cannot deny this is that we have the relation v=x/t. He says it is incorrect to do this because the instrument was a clock not a velocity measuring meter, such as a "Doppler shift velocity meter". It may come as a surprise to Peter Cheeseman, but any instrument that would measure frequencies incorporates a clock that measures intervals of times. This is because a frequency is defined as a number of periods per unit of time. Is Peter Cheeseman not aware that for about two decades the internationally agreed upon definition of a second is based upon the frequency of a particular transition of cesium atoms? Since Peter Cheeseman may still argue it matters how one obtains v2 < v < v1, let us suppose that the robot is equipped with redundent sensors. That is to say, it is equipped with a clock that measures the interval of time t that the object takes to cover the distance x, and it is also equipped with a "Doppler shift velocity meter" which, since it seems to matter to Peter Cheeseman, has a linear scale. One reason for having redundant sensors is that we could check whether their outputs are consistent or not. Let us immagine that the sensors give not only consistent outputs, but in fact identical outputs. That is to say, the clock gives t1 < t < t2, and the "velocity meter" gives precisely v2 < v < v1, where v1=x/t1 and v2=x/t2. What does Peter Cheeseman instruct the robot to do? If I have correctly understood the above quotation from his answer to Gerard, the robot should say on the basis of its "clock sensor" that:

P(t)dt=dt/(t2-t1), for t1 < t <t2,

and on the basis of its "velocity meter" that

P(v)dv=dv/(v1-v2), for v2 < v < v1.

Since we have v=x/t and v1=x/t1 and v2=x/t2, these two probability distributions are inconsistent. They are inconsistent because I have shown that the distribution P(t)dt=dt/(t2-t1) transforms into the distribution P(v)dv=dv/((v1-v2)*v**2), and not the distribution P(v)dv=dv/(v1-v2) that Peter Cheeseman says is the correct Bayesian thing to do on the basis of the "velocity meter".

It is precisely to deal with situations such as the one I have just considered, with the two different meters that give precisely the same answers from a physical point of view, that Jaynes devised his transformation group methods based upon his Desideratum of Consitency. Jaynes would insist that whatever one assigns to P(t)dt on the basis of the "clock meter" transforms into the P(v)dv that one assigns on the basis of the "velocity meter", and vice versa, when what these two meters give is identical.

I hope to have convinced Peter Cheeseman that what he proposes does indeed lead to a contradiction. It is amusing to me that Peter Cheeseman did not know what is claimed to be the correct Bayesian answer to this problem, since it was published in 1939 by Jeffreys. In fact the distribution that one must assign to t, P(t)dt, and also to v, P(v)dv, in order for the two answers to be consitent, is known among Bayesians as the Jeffreys distribution. It is:

P(t)dt=dt/(t*ln(t2/t1)), for t1 < t < t2,

which transforms into

P(v)dv=dv/(v*ln(v1/v2)), for v2 < v < v1,

and conversely.

Jeffreys invented a rule to decide when one should assign a Laplace prior, i.e. a uniform one, and when one should assign a "Jeffreys prior", i.e. one proportional to dx/x where x is the parameter. This rule is: if the parameter can take on positive and negative values, for instance an angular position with respect to an origin, then the Laplace prior should be used, but if the parameter can only take on positive values, such as an interval of time or a density or a velocity , then one should assign the Jeffreys prior. Jaynes verbalizes this in a different manner, but obtains the same results, in most instances, with his Desideratum of Consistency. He says that the Laplace prior is the non-informative prior for a position parameter, but the Jeffreys prior is the non-informative prior for a scale parameter.

To conclude this rather lengthy note, I will state a problem that cannot be solved by Jaynes methods. This problem is known to Jaynes, and should be known to Peter Cheeseman since it is mentioned in the article of Jaynes that he sent with his answer to Gerard. Jaynes conjectured in this article that this problem was really "ill posed". Unfortunately for Peter Cheeseman's Challenge it is one that a robot could be called upon to solve. This problem is known as Von Misses Water and Wine problem. Let me show that a robot could be called upon to solve this problem. A robot is equipped with a sensor that analyzes some binary mixtures, for instance this sensor is a mass spectrograph or a separation collumn, the robot must make a decision based upon the concentration of the binary mixture. The sensor samples a given amount of the mixture and tells the robot that the ratio of the amount, a, of one component of the mixture to the amount, b, of the other component is between 1 and 2. Since this tells the robot that the inverse ratio, b/a, is between 1/2 and 1, what is the probability distribution that the robot should assign to both of these ratios in order not to have a contradiction? It turns out that if one applies Jeffreys rule no contradiction results. This I presume was well known to Jaynes when he wrote his paper. However, Jaynes refused to use the Jeffreys prior, because he knows that the Jeffreys prior can only be used for a scale parameter, and these ratios are not scale parameters since they are dimmensionless. I have proposed a new theory of probabilities that produces very precisely the answers that Jaynes obtains with his transformation group arguments, when it can be applied, and his Maximum Entropy method, when it is appropriate to use it. But this new theory can also solve Von Misses Water and Wine Problem, and address the problem of the probability amplitudes of quantum mechanics. If one were to apply what Peter Cheeseman sugests one should do, to which of the two ratios, a/b or b/a, would he apply the uniform distribution? The sensor does not measure either ratios, it obtains a and b and then, depending upon the switch setting on the instrument, makes the division a/b or b/a, which it communicates to the robot. The answer that the robot provides cannot be a function of the switch setting on the sensor, because in order to provide some redundency I will insist that the robot be given both the ratios a/b and b/a, of course it will be told which is the ratio a/b and which is the ratio b/a.

October 15, 1985

C. R. Weisbin

Cheeseman's Challenge (continue)

Francis pointed out that I had misinterpreted his solution to the Bayesian Prior Problem. Consequently,I have revised my letter of August 17 and removed the paragraph describing Francis' solution (his notes to you explain his approach). I have also replaced "most probable estimate" by best estimate which, as pointed out by Cheeseman is a more correct description.

After rereading Cheeseman's response, The Well Posed Problem of Jaynes (particularly the Appendix of that paper) and Francis's memos to you of 9/30 and 10/7 I am more convinced than ever that some interpretations of Bayesian non-informative prior may lead to contradiction.

I think that there is nothing vague, ambiguous or incomplete in my description of the problem illustrating the Noninformative Prior Paradox. Yet Peter and Francis propose rather different "solutions" to this problem. I am more in agreement with Francis's approach; but he uses theory that goes beyond "Bayesian methods augmented with maximum entropy."

Gerard de Saussure Gérard

GdS:sar

Notes on the CESAR workshop: Planning and Sensing for Autonomous Navigation

August 18-19, 1985

Ralph L. Hollis
IBM Thomas J. Watson Research Center

The meeting was generally an interesting one, but there was not enough time for informal discussion among the participants. I felt that the theme topics were a bit too broad, and needed more focus. We seemed to develop answers to the various questions at only the highest most "meta" level, as if we actually thought we could do a top-down design for an intelligent robot system (in two days!). As a result, we frequently drifted off into philosophical considerations.

I think the most useful presentations were those which were most concrete, e.g. those of Elfes and Harmon. I would like to see in future workshops more discussion of existing or near-term planned mobile robot systems, highlighting the areas of difficulty. A great deal of practical experience has been gained over the years by those of us who have actually built mobile robot systems, and I think it would be useful to share some of this in a group. By looking at a number of different attempted solutions to problems, e.g. sensing in a mobile environment, or path finding, we can then hope to extract generalizations.

LIST OF PARTICIPANTS

LIST OF PARTICIPANTS

"PLANNING AND SENSING FOR AUTONOMOUS NAVIGATION"
August 18-19, 1985, UCLA

Dr. Jacob Barhen	ORNL, CESAR
Dr. Peter Cheeseman	NASA-AMES
Professor James Crowley	Carnegie Mellon University
Dr. G. deSaussure	ORNL, CESAR
Dr. Martin Dudziak	Martin Marietta Aerospace
Dr. Alberto Elfes	Carnegie Mellon University
Dr. W. R. Hamel	ORNL, CESAR
Dr. Scott Harmon	Naval Ocean Systems Center
Dr. Ewald Heer	Heer Associates
Dr. Steven Holland	GM Research Laboratories
Dr. Ralph Hollis	IBM Research
Professor Ramesh Jain	University of Michigan
Dr. Leslie Kaelbling	Stanford Research Institute
Dr. Yutaka Kanayama	Stanford University
Dr. David Keirey	Hughes Aircraft
Professor Alexander Meystel	Drexel University
Professor David Miller	Virginia Polytechnic Institute
Dr. E. M. Oblow	ORNL, CESAR
Dr. Alex Parodi	FMC Corporation
Dr. Alex Pentland	Stanford Research Institute
Professor K. N. Reid	Oklahoma State University
Dr. Bill Richard	Sandia National Laboratory
Dr. Stan Rosenschein	Stanford Research Institute
Dr. Robert Tilove	GM Research Laboratories

RESUMES

RESUMES

1. PERSONAL DATA

Name	:	Jacob BARHEN
Date & Place of Birth:		October 15, 1948, Sofia, Bulgaria
Marital Status	:	Married, one daughter
Citizenships	:	Israeli and French
Visa Status	:	Permanent Resident (U.S.A.)
Home Address	:	115 North Seneca, Oak Ridge, TN 37830 phone: 615-483-8693
Work Address	:	ORNL, P.O. Box X, Oak Ridge, TN 37831 phone: 615-574-6162 (FTS: 624-6162)

2. DIPLOMAS

November 1978	:	Doctor of Science Technion - Israel Institute of Technology
November 1975	:	Master of Science in Nuclear Engineering Technion - Israel Institute of Technology
June 1970	:	Maitrise es Sciences in Physics University of Paris, France
September 1968	:	Diplome Universitaire d'Etudes Scientifiques University of Paris, France

3. EDUCATIONAL BACKGROUND

10/75 - 9/78	:	Nuclear Engineering Technion - I.I.T., Israel
08/74 - 9/75	:	Applied Mathematics, Weizman Institute of Science, Israel
04/73 - 6/74	:	Academic Reserve Division, Israel Defense Forces
03/72 - 3/73	:	Nuclear Engineering, Technion - I.I.T., Israel
10/68 - 6/70	:	Physics, University of Paris, France
10/66 - 6/68	:	Mathematics, University of Paris, France

4. CHRONOLOGY OF EMPLOYMENT

06/85 - Date	: Senior Research Staff, Engineering Physics & Mathematics Division, Oak Ridge National Laboratory
04/82 - 05/85	: Research Staff II, Engineering Physics & Mathematics Division, Oak Ridge National Laboratory
09/81 - 03/82	: Research Staff I, Engineering Physics Division, Oak Ridge National Laboratory
10/78 - 08/81	: Research Associate III, Engineering Physics Division, Oak Ridge National Laboratory
10/75 - 09/78	: Research Assistant, Department of Nuclear Engineering Technion - Israel Institute of Technology

5. CHRONOLOGY OF POSITIONS HELD

06/85 - Date	: Head, Machine Intelligence and Advanced Computer Systems Group
07/83 - 05/85	: Group Leader, Advanced Energy Systems Group: administrative and technical supervision of multiple projects
11/80 - 06/83	: Project Leader, Energy Systems Analysis Group: administrative and technical supervision of a project involving 8 FTE professionals
01/80 - 10/80	: Task Leader, Reactor Methods and Data Development Group

6. MEMBERSHIP IN SCIENTIFIC ASSOCIATIONS

American Association for Artificial Intelligence	: member
American Defense Preparedness Association	: member
American Nuclear Society	: member
IEEE	: member
Robotics International	: senior member
Israel Nuclear Society	: member

7. AWARDS

Best paper award, American Nuclear Society, Reactor Physics Division, November 1981.

Technion's Board of Governors achievement award scholarships, in 1977, 1976, and 1975. The A. Rubin special award scholarship in 1972.

8. RECENT PROFESSIONAL ACTIVITIES

Chairman and organizer, sessions on "Advances in Concurrent Computation," 1985 International Computers in Engineering Conference, Boston, MA (August 4-8, 1985).

Moderator and organizer, panel on "Future Directions in Concurrent Computation," 1985 International Computers in Engineering Conference, Boston, MA (August 4-8, 1985).

Member, program committee, IEEE Second Annual Conference on Artificial Intelligence Applications, Miami-Beach, FL (December 11-13, 1985).

9. RECENTLY GENERATED FUNDING

Virtual-Time Systems for Distributed Computing; source: AFOSR/WPAFB, FY'(85), $200K (principal investigator).

Theoretical Foundations for Concurrent Computation; source: ORNL Director discretionary funds, FY'85, $60K (principal investigator).

Intelligent Control Systems (Center for Engineering Systems Advanced Research); source: DOE/BES/ERP, FY'(83), $2M (principal investigator).

Development of Strategic Planning Capabilities for Liquid and Gaseous Fuels Supply; source: DOE/OPE, FY'(81→), $1.8M (principal investigator).

10. OTHER ACTIVITIES AND HOBBIES

Board of Directors, Jewish Congregation of Oak Ridge

Archeology: Mayas (classic period); Incas (Tiahuanaco)

Travel, swimming, backgammon

11. LANGUAGES

English, French, Spanish	:	fluent
Hebrew, German	:	good

12. CURRENT RESEARCH PURSUIT AND OBJECTIVES

Machine Intelligence. Provided the key technical leadership which resulted in the establishment at ORNL of CESAR, the Center for Engineering Systems Advanced Research. CESAR's long-term mission is to perform basic research in intelligent control systems with special emphasis on machine intelligence and advanced computer architectures for autonomous robots operating in hostile environments.

Autonomous robots are generally composed of a variety of asynchronously controlled components such as manipulator arms, electro-optical sensors, sonars, navigation controllers, etc. In order to take advantage of the distributed nature of the associated robotic processes, we are currently developing a Virtual-Time Robot Operating System (ROS/VT), which is intended to provide a generalized frame-work for implementing machine intelligence. Recent advances in VLSI technology, which have resulted in the successful development of powerful hypercube ensembles for concurrent computation - a major breakthrough in the area of supercomputing - have provided us with a strong incentive to configure the "brain" of our intelligent robot as a homogeneous hypercube of appropriate dimensionality. It should be emphasized that an essential step to ensure the success of this approach to machine intelligence is the development of adequate algorithms for concurrent computation.

Algorithms for Concurrent Computation. This task is far more difficult in the framework of intelligent robotics, than for the usually demanding computations encountered in the classical fields of science and engineering. In the latter (including, for example, matrix, grid or finite element formulations) the algorithm structure is so regular, that the corresponding processes can be mapped directly onto the hardware topology. For intelligent machine applications, the structures of typical processes are both irregular and involve nonlocal communications. Even for "static" processes (those modeling, for example, the arm's inverse dynamics equations), this requires that the mapping of the process structure (task graph) onto the hypercube be computed prior to execution. This endeavor is extremely difficult, particularly when precedence constraints are involved. A prototype algorithm, ROSES (ROS Expert Scheduler), is currently being tested, and shows excellent promise for controlling the real-time solution of, e.g., the inverse dynamics equations.

Further down the road, if we allow for the process structure to evolve dynamically, as may be required for robots operating in unstructured environments, considerable complications arise. These include the development of appropriate methodologies for real-time mapping of task graphs onto the machine's topology, the capability for processes to spawn or annihilate other processes, and more generally for the operating system to be capable of "reasoning in time", i.e., handling correctly a set of artificial time scales in a distributed computing environment.

Robotics. Current research interests are directed toward the development of efficient mathematical models for solving the dynamics of flexible arms.

13. QUALIFICATIONS, SKILLS, AND PAST RESEARCH EXPERIENCE

Light Water Reactor Physics. Was trained at the Technion - Israel Institute of Technology under Professor W. Rothenstein, a world-recognized leader in reactor physics. While at the Technion, was actively involved in the development of advanced theoretical methods for Light Water Reactor benchmark calculations. In particular, authored a sophisticated new methodology for the computation of accurate resonance reaction rates in heterogeneous reactor lattices. This methodology is embodied in a unique computer code currently in use both in the U.S. (Electric Power Research Institute, Brookhaven National Laboratory, ORNL, etc.) and Canada (Chalk River National Laboratory).

Highly specialized in the development of benchmark procedures for reactor core analysis. Principal investigator on a number of projects sponsored by the Electric Power Research Institute. Additional experience in core fuel management and optimization, and in radiation transport.

Fast Breeder Reactor Physics. Expertise in neutron cross sections generation techniques and Bondarenko formalisms, particularly in the unresolved energy region and for the shielding of scattering matrices. Coauthored the development of the first major United States fine group cross section library based on ENDF/B-V data. The resulting computer codes and data are used by major reactor design organizations throughout the country.

Sensitivity and Uncertainty Analysis. Demonstrated that current state-of-the-art uncertainty analysis techniques for reactor safety problems exhibit major shortcommings. Developed a sophisticated new methodology for uncertainty analysis of time-dependent nonlinear systems. When applied to the analysis of severe thermal-hydraulic transients (e.g., blowdown accidents), the new methodology, which incorporates in-bundle measurements, achieved, for the first time in reactor safety, a systematic reduction of uncertainties. It is expected to have significant impact on reducing margins of safety related costs.

Large-Scale Systems. Responsible (management and scientific leadership) for a DOE/FE program on developing strategic planning capabilities for liquid and gaseous fuels supply. The program aims at designing models of regional, national and international energy-economic markets and their linkage to the rest of the economy. The resulting models are applied to evaluate the impact of specific public policy or private sector investments in technology research and development alternatives [e.g., Enhanced Oil Recovery R+D]. Quantitative estimates of costs, benefits and associated uncertainties require development of advanced sensitivity and uncertainty analysis techniques for large-scale time-dependent systems. An automated decision analysis methodology based on sensitivity theory is also being developed, to provide operational support for the Division of Policy and Strategic Planning, DOE/Fossil Energy.

14. REFEREE

Referee, Proposal Reviews, National Science Foundation
Referee, Proposal Reviews, DOE/Office of Basic Energy Sciences
Referee, Nuclear Science and Engineering
Referee, American Control Conference

15. STUDENT SUPERVISION

In the framework of the Oak Ridge Associated Universities Science Semester at ORNL, has supervised 11 students since 1980 [1980 (1); 1981 (3); 1982 (2); 1983 (3); 1984 (2)].

16. UNIVERSITY TEACHING

Teach graduate course "Introduction to Concurrent Computation" at the University of Tennessee (both EE and CS Departments)

James L. Crowley is a Research Scientist with the Robotics Institute where he is Director of the Household Robot Laboratory. He is Principal Investigator of the project *Navigation and World Modeling for a Mobile Household Robot* and Co-Principal Investigator on two vision projects: *Multi-Resolution Representation and Matching of Shape* and *Hybrid Syntactic/Statistical Pattern Recognition*. Dr. Crowley received his B.S. in Electrical Engineering from Southern Methodist University in 1975, and his M.S. and Ph.D. degrees in Electrical Engineering from Carnegie-Mellon University in 1977 and 1982, respectively.

Since joining the Robotics Institute in May 1980, he has completed three projects dealing with measuring, representing, and matching three-dimensional shapes for industrial applications: The Four-Camera Light-Stripe Sensor, the Panto-Scribe Cross Section Measurement System, and the Multi-Resolution Chamfer Matching System. The Panto-Scribe System and the Multi-Resolution Chamfer Matching System are installed in a Westinghouse Turbine Components Factory in Winston-Salem, North Carolina.

Dr. Crowley's research interests include dynamic 3-D scene analysis, representation and matching of shape, and mobile robot navigation. For his Ph.D. dissertation, Dr. Crowley developed a representation for shape based on peaks and ridges in the Difference of Low-Pass (D.O.L.P.) Transform. This representation provides the basis for development of a 3-D scene analysis system, for a gray-scale vision system, and for applications in industrial inspection.

Gerard de Saussure
P.O. Box X, Bldg. 6010
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831

Employee Number:	06226
Home address:	100 Windham Rd. Oak Ridge, TN 37830
Home Phone Number:	483-8796
Citizenship:	USA
Clearance:	Current Q
Date of Birth:	11/22/24
Sex:	Male

PROFESSIONAL EXPERIENCE

8/55 to Present — Senior Research Staff Member 1
Engineering Physics and Mathematics Division, Oak Ridge National Laboratory. Long experience in broad area of measurement and interpretation of neutron cross sections required for nuclear reactor design and assessment.

PROFESSIONAL AND ACADEMIC HONORS

1975 "Best Paper Award" by the Reactor Physics Division of the American Nuclear Society for the years 1973-74.

1981 Reactor Physics Division "Best Paper Award" at the 1981 Winter Meeting.

PROFESSIONAL SOCIETIES

Member - American Physical Society
Fellow - American Nuclear Society
Officer/Committee - American Nuclear Society

SPECIAL LABORATORY/DOE/UCC ASSIGNMENTS

00/71	00/76	Member, Cross Section Working Evaluation Group (CSEWG)
00/78	00/78	Invited for 3-month assignment at the Centre D'Etudes de Bruyere-le-Chatel

00/75	00/75	Invited for 3-month assignment at the Centre D'Etudes de Bruyere-le-Chatel
00/70	00/73	Member, National Cross Section Advisory Group (NCSAC) subcommittee on fission
09/63	09/64	Mutual exchange assignment, Department de Physique, Centre D'Etudes Nucleaires (CEN), Saclay, France
00/00	00/00	Six weeks consultant to LASL (77).

OTHER SPECIAL ASSIGNMENTS

00/80	Present	Member, Thesis Committee, Ph.D. Thesis, University of Tennessee
00/80	Present	Honorary Professor, Nuclear Engineering Department, University of Tennessee
06/79	06/83	Chairman, International Meetings Subcommittee of National Program Committee, Am. Nucl. Soc.
06/77	06/83	Member, Steering Subcommittee of National Program Committee, Am. Nucl. Soc.
06/77	06/83	Member, National Program Committee, American Nuclear Society
00/71	Present	Referee, Nuclear Science and Engineering and the Physical Review
00/65	06/80	Member, Program Committee, Reactor Physics Division, Am. Nucl. Soc.
00/73	00/77	Chairman, Program Committee, Reactor Physics Division, Am. Nucl. Soc.
00/74	00/76	Member, Executive Committee of the Reactor Physics Division of the American Nuclear Society
09/84		Member, Program Committee and Session Organizer, Topical Meeting on Reactor Physics and Shielding, American Nuclear Society, September 17-19, 1984.
05/85		Member, General Program Advisory Committee, International Conference on Nuclear Data for Basic and Applied Science, May 13-17, 1985
05/85		Member, Technical Program Committee, Fifth Pacific Basin Nuclear Conference, American Nuclear Society, May 1985

ACADEMIC EDUCATION

1954 Doctorate Degree
Massachusetts Institute of Tech., Cambridge, Mass.
Major: Experimental Physics

Thesis Topic or Title:
Elastic Photoproduction of 11 Degrees Mesons in Helium and (VN) Reaction on Helium at High Energies

Foreign Languages
French
German
Italian

Alberto Elfes

October, 1985

The Robotics Institute
Carnegie-Mellon University
Pittsburgh, PA 15213
(412) 578-8805

7070 Forward Avenue, Apt. 907
Pittsburgh, PA 15217
(412) 521-5423

Current Position

Ph.D. Candidate, Electrical and Computer Engineering Department and Robotics Institute, Carnegie-Mellon University, since March 1982.
Areas of Research: Artificial Intelligence, Computer Vision and Robotics.

Research Assistant, Mobile Robot Laboratory, The Robotics Institute, Carnegie-Mellon University, since May 1982.

Education

Ph.D., Electrical and Computer Engineering Department and Robotics Institute, Carnegie-Mellon University. Expected completion date: August 1986.
Thesis Title: *A Software Architecture for Mobile Robots.* Advisors: Hans P. Moravec and Sarosh N. Talukdar.

M.Sc. in Computer Science, Instituto Tecnológico de Aeronáutica, Brazil, November 1980.
Thesis Title: *Knowledge Representation and Decision Process Description in Medical Diagnosis.*

B.Eng. in Electronics Engineering, Instituto Tecnológico de Aeronáutica, Brazil, December 1975.

Current Research Activities

Development of a Sonar-Based Mapping and Navigation System for mobile robots. This includes processing of sonar range data to build occupancy maps, object extraction, path-planning and locomotion. The system has been tested in indoor and outdoor environments. The research is being conducted for the Mobile Robot Lab and the Autonomous Land Vehicle (ALV) project.

Design and implementation of a software architecture for mobile robots. Aspect of sensor integration, real-world modelling, multi-level planning and problem-solving and distributed and cooperative computing are being analysed. This research is being done for the Mobile Robot Lab.

Development of a Distributed Problem-Solving framework. A general framework was developed and a set of primitives was built to allow the implementation of problem-solving systems that operate by having multiple experts cooperating in the solution of a problem. This research is being sponsored by the Mobile Robot Lab and the Design Research Center, CMU.

Research Interests

Artificial Intelligence, Computer Vision, Robotics

Mobile Robots: Sensory Processing, Software Architectures, Planning and Problem-Solving, Sensor Integration

Frameworks for Distributed and Cooperative Problem-Solving

Advanced Computer Architectures and Languages for Parallel and Distributed Processing

VLSI and Processor Design Automation

Professional Experience

Positions Held

Carnegie-Mellon University:

Research Assistant, Mobile Robot Laboratory, The Robotics Institute, Carnegie-Mellon University, since May 1982.

Research Assistant, Design Research Center, Carnegie-Mellon University, Summer 1984.

Research Consultant for Denning Mobile Robotics, Inc., Woburn, MA. Since 1984.

Computer Science Department, Instituto Tecnológico de Aeronáutica:

Assistant Professor, Computer Science Department - Instituto Tecnológico de Aeronáutica (ITA), Brazil, since January 1981. On leave.
Taught Graduate level courses in Artificial Intelligence, Compiler Construction and Computer Architecture.
Research work in Computer-Based Medical Diagnosis Systems, Knowledge Representation Methods and Expert Decision Process Description.

Assistant Head, Computer Science Department, ITA, 1981 - 1982.

Member of the ITA Faculty Senate, 1978 - 1982.

Computer Science Program Coordinator, Computer Science Department, ITA, 1979 - 1981.
Planned Undergraduate and Graduate Curricula in Computer Science and Engineering. Supervised the departmental teaching activities.

Head, Computation Center, ITA, 1978 - 1979.
Supervised the operation of the computing facilities at ITA.
Member of the Computer Selection Committee of the Brazilian Aerospace Technical Center (CTA).

Research Advisor for Senior Research Projects, Computer Science Department, ITA, 1977 - 1981.
Research projects were developed in the areas of compiler construction, pattern recognition, medical expert systems, graphics systems, chess programs and knowledge representation methods.

Teaching Assistant, Computer Science Department, ITA, 1976 - 1980.
Taught Undergraduate level courses in Computer Organization, Programming and Numerical Methods.
Research work in Pattern Recognition and Interactive Graphics Systems.

Undergraduate Employment, Instituto Tecnológico de Aeronáutica:

Research Assistant, National Space Research Institute (INPE), Brazil, 1975.
Used pattern recognition methods to implemented a Handwritten Character Recognition System.
Developed Image Processing software and User Interface for a Satellite Image Processing System.

Research Trainee, Computer Science Department, ITA, 1974.
Fellowship from the Research Support Foundation of São Paulo (FAPESP).
Development of an Interactive Graphics System for Exploratory Data Analysis.

Research Trainee, Computer Science Department, ITA, 1973.
Development of Systems Software for Computer Science Education.

Other Activities

Invited Paper: Workshop on Planning and Sensing for Autonomous Navigation, sponsored by Oak Ridge National Laboratory, UCLA, Los Angeles, August 1985.

Invited Speaker: *Robotics - Principles and Applications*, talk delivered at the IBM Scientific Center, Brasília, Brazil, July 1983.

Short Course: *Introduction to Robotics*, one-week graduate-level tutorial, Electronics Engineering Department, ITA, July 1983.

Invited Speaker: *Artificial Intelligence and Computer-Based Medical Diagnosis*, talk delivered at the Mathematics, Statistics and Computer Science Institute, University of Campinas, Brazil, June 1981.

Invited Speaker: *Knowledge Representation and Decision Process Description in Medical Diagnosis*, talk delivered at the Heart Institute, University Hospital, University of São Paulo, Brazil, April 1981.

Invited Speaker: *Artificial Intelligence in Medical Diagnosis*, talk delivered at the Symposium on Artificial Intelligence, sponsored by the Brazilian Society for the Progress of Science (SBPC), Brazil, July 1980.

Invited Speaker: *Medical Diagnosis by Computer*, talk delivered at the CECE - IBM, Brazil, May 1980.

Invited Speaker: *Medical Diagnosis by Computer*, talk delivered at the Institute for Space Research (INPE), Brazil, April 1980.

Professional Societies

Association for Computing Machinery (ACM), Student Member
Institute of Electrical and Electronic Engineers (IEEE), Student Member

Personal Data

Born:	September 09, 1953, in Maceió, Brazil
Citizenship:	Brazilian
Marital Status:	Married, two children

Languages Spoken

Fluent in Portuguese, German and English.
Working knowledge of Spanish and French.

RESUME

NAME: William Ross Hamel

ADDRESS: 328 Dominion Circle
Farragut, TN 37922

PHONE: (615)-966-9337

PERSONAL DATA: Married
Children: Luke 8, Alison 5
Height: 5'-8"
Weight: 165 lbs.
Age: 36
Physical Condition: Excellent
U. S. Citizen

PROFESSIONAL EXPERIENCE:

(1). 12/81-Present Remote Control Engineering Task Leader, Consolidated Fuel Reprocessing Program, Oak Ridge National Laboratory (ORNL); Oak Ridge, TN.

Responsiblities: Development of advanced remote handling systems for breeder reprocessing applications including advanced servomanipulator development. Involves definition and justification of technical scope, and execution of program plan within budget and schedule.

(2). 12/78-12/81 Special Assignment and Group Leader of Mathematical Methods Group, in Measurements and Controls Engineering Section, Instrumentation and Controls Division, ORNL.

Responsiblilites: Multiple activities including research in Coriolis mass flowrate measurements, study of remote systems technology in Light Water Reactor maintenance for occupational radiation exposure reduction, coal preparation process modeling and automation analysis, and robotics. Developed proposals and performed "sales" presentations.

(3). 3/77-12/78 Instrumentation and Control Manager, Consolidated Fuel Reprocessing Program, ORNL.

Responsiblities: Planning, coordination, and execution of instrumentation and control related activities associated with breeder fuel reprocessing technology development.

Included both basic research and development and large project support and monitoring.

(4). 7/72-3/77 Development Engineer, Process Systems Group, Measurements and Controls Engineering, Instrumentation and Controls Division, ORNL.

Responsibilities: Process instrumentation engineering design and field support in a wide range of applications, dynamic modeling and analysis of process control systems, digital and hybrid computer simulation.

(5). 7/70-7/72 Project Engineer, Control Technology and Simulation Group, Measurement and Controls Technology, South Charleston Technology Center, Union Carbide Corporation, Chemicals and Plastics Division, South Charleston, WV.

Responsibilities: Process control systems design. Analog, hybrid, and digital computer simulation of "difficult" process control systems. Organized corporate process control technology development program.

(6). 1/69-7/70 Member of Professional Staff, TRW Systems Group, Houston Operations, Houston, TX.

Responsibilities: (1). Digital computer simulation of heat transfer in cryogenic storage vessels for Earth/Mars mission spacecraft. (2). Apollo Lunar Module digital autopilot flight software analysis, verification, and performance evaluation using bit-by-bit digital simulation. Postflight data analysis for Apollo Missions 8, 9, 10, 11, and 12.

(7). 9/67-1/69 Research Assistant, School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK.

EDUCATION:

1981 PhD	University of Tennessee Mechanical Engineering Dissertation: "Analysis of a Cantilever Coriolis Mass Flowmeter Concept"
1969 MSME	Oklahoma State University Mechanical Engineering

1967 BSME West Virginia University
Mechanical Engineering

HONORS AND SOCIETIES:

Tau Beta Pi, Sigma Xi, Phi Kappa Phi
NSF Summer Teaching Fellowship (1968)
FMC Corporation Tuition Scholarship (1966)

Institute of Electrical and Electronics Engineers
Americal Society of Mechanical Engineers

PATENTS:

(1). Disclosure: (1981) A Cantilever Coriolis Mass Flowmeter

(2). Granted: (1978) Automatic Inspection System for Nuclear Fuel Pellets or Rods, R. A. Bradley, W. R. Hamel, W. H. Miller, and J. D. Sease.

PUBLICATIONS:

(1). Hamel, W., "Analysis of a Cantilever Coriolis Mass Flowmeter Concept," Ph.D. Dissertation, University of Tennessee, 1981.

(2). Hamel, W. and Reid, B., "Remote Systems Technology, Occupational Radiation Exposure and Light Water Reactors," Workshop to Delineate the Economic, Technical, and Policy Issues for Remote Maintenance in Energy Systems, University of Florida, 1981.

(3). Canright, G., Allgood, G., Brown, C. and Hamel, W., "Dynamic Modeling and Control Analysis of Froth Flotation and Clean Coal Filtration as Applied to Coal Beneficiation," ORNL/TM-8015, 1981.

(4). Canright, et. al, "Transient Modeling of Froth Flotation and Vacuum Filtration Processes," Symposium on Instrumentation and Control for Fossil Energy Processes, San Francisco, 1981. (Republished ISA Trans., September 1982)

(5). Canright, et. al, "Dynamic Simulation of the Coal Froth Flotation Process of Control Analysis," Symposium on Instrumentation and Control for Fossil Energy Processes, San Francisco, 1981.

(6). Hamel, W., Jansen, J., and Bradley, N., "Instrumentation and Control Development and Design Philosophy for Advanced Fuel Recycle," ORNL/TM-6393, 1978.

(7). Hamel, W. and McDuffee, T., "Value Impact of Vault Automation in Special Nuclear Material Storage," ORNL/NUREG-33, 1978.

(8). Hamel, W., et. al, "Design Criteria for Vault Automation in Special Nuclear Material Storage," ORNL/NUREG-44, 1978.

(9). Hamel, W. and Shappert, L., "Desirability and Feasibility of Vault Automation in Special Nuclear Materials Storage," ORNL/NUREG-20, 1977.

(10). Sebesta, H., Hamel, W. and Leight, J., "Conditions for Optimal Control of Time Delay Systems Subject to Inequality Constraints", IEEE 12-th Midwest Symposium on Circuit Theory, University of Texas, 1969.

CAREER INTERESTS:

Line management in measurement and controls engineering, research, and development. Desire a role in which there is opportunity for a mixture of both technical and managerial responsiblities. Current interest in robotics technology.

Ewald Heer
5329 Crown Avenue
La Canada, California 91011
(213) 790-3799

EDUCATION

University Hanover, Germany, 1962-1964.
- Doctor of Engineering Sciences (Magna Cum Laude) (1964).

Columbia University, New York, New York, 1959-1962.
- Master of Science (1960), Major in Engineering Mechanics.
- C. E. Professional Degree (1962), Major in Engineering Mechanics.
  Certificate for "Outstanding Achievement as a Graduate Student."

New York University, 1959. Theoretical Physics (two courses)

City University, New York, 1957-1959.
- Bachelor of Science (1959), Major in Physics.

School of Architectural Engineering, Hamburg, Germany. 1948-1953.
- Architectural Engineer (1953).

California Institute of Technology Courses:
  Software/Hardware Project Management (1980)
  Procurement Management (1967)
  Management Techniques (1967)
  Motivation by Results (1967)

MEMBERSHIP IN SOCIETIES

SIGMA XI Honorary Society
OMEGA RHO International Honor Society
New York Academy of Sciences
American Association for the Advancement of Science
American Society of Civil Engineers
American Society of Mechanical Engineers
Institute of Electrical and Electronic Engineers
American Institute of Aeronautics and Astronautics
Operations Research Society of America
International Federation for the Theory of Machines and Mechanisms
American Management Association
Society of Manufacturing Engineers
Robot Institute of America
Institute of Industrial Engineers

LISTED

American Men and Women of Science
International Scholars Directory
International Who's Who of Intellectuals
Who's Who in the West
Who's Who in Finance and Industry
Who's Who in the World
Who's Who in Engineering
Men of Achievement

REGISTRATION
Licensed Professional Engineer, State of New York since 1962
Licensed Professional Engineer, State of California since 1971

UNIVERSITY APPOINTMENTS

School of Architectural Engineering, Hamburg, 1963. Junior course: "College Physics" (in German).

Pennsylvania State University, 1965. Seminar course: "Basic and Random Vibration."

University of Southern California, 1968. Senior course: "Aerospace Structures."

University of California, Los Angeles, 1969. Planned and taught graduate course: "Engineering Analysis and Design: Safety Factor to Modern Statistics."

University of Southern California, since 1973.
Adjunct Professor of Industrial and Systems Engineering.

- Courses taught:

1. Mathematical Methods of Operations Research (graduate course)
2. Operations Research (graduate course)
3. Network analysis and Planning (graduate course)
4. Engineering Economics (senior undergraduate course)
5. Directed Research (graduate level)
6. Supervision of graduate students at masters, engineering, and doctorate levels
7. Value and Decision Theory (graduate course)

- Contract Research for NASA.

- Chairman of Steering Committee on Robotics Education.

PROFESSIONAL COMMITTEES

- Member of the ASCE Engineering Education Committee, Los Angeles Section, 1969.
- Member of the AIAA Professional Member Education Committee, 1970 to 1972.
- Member of the NASA Committee on Teleoperators and Robots, and Chairman of the Panel on Mobility Units and Manipulators, 1970.
- Member of the NASA Committee on Remotely Manned Systems and Extravehicular Activities (RMS/EVA), and Chairman of the Lunar and Planetary Subcommittee, 1971 to 1975.
- Member of the NRC Subcommittee on Vision since 1974.
- Organized NASA Study Group on Machine Intelligence and Robotics, and served as Executive Secretary under the chairmanship of Dr. Carl Sagan. The Study Group's objective was to assess the NASA Computer Science and Automation Program and make recommendations for long range technology developments.
- Chairman, ASME Technical Committee on Rotobics and Manipulators since 1979.

PROFESSIONAL COMMITTEES (Cont'd)

- Executive Secretary, Executive Committee, ASME Computer Engineering Division, 1980 to 1981
- Vice Chairman, ASME Computer Engineering Division, 1981 to 1982
- Chairman, ASME Computer Engineering Division, 1982 to 1983.
- General Chairman, 1984 ASME International Computers in Engineering Conference and Exhibit, August 1984.

PROFFESIONAL CONFERENCES

- Organized and chaired the "First National Conference on Remotely Manned Systems" held at the California Institute of Technology, September 13-15, 1972.
- Chairman, First International Symposium on Theory and Practice of Robots and Manipulators, Session on Kinematics and Dynamics, Udine, Italy, 1973.
- Chairman, 17th Annual Human Factors Society Convention, Session on Remote Control Applications, New York, 1974.
- Organized and Chaired the "Second Conference on Remotely Manned Systems -- Technology and Applications," held at the University of Southern California, June 11-13, 1975.
- Chairman, Joint Automatic Control Conference, Session on Man-Machine Systems, San Francisco, May, 1977.
  Session Organizer AIAA/NASA Conference on "Smart Sensors, NASA Langley Research Center, Hampton, Virginia, November 14-16, 1978.
- Chairman Annual Rocky Mountain Guidance and Control Conference, Autonomous Systems Session, Keystone, Colorado, February 24-28, 1979.
- Chairman, International Computer Technology Conference, Robots and Manipulators Session, ASME CENTURY 2, San Francisco, California, August 12-15, 1980.
- Organizer and Program Chairman of the Conference on Automated Decision Making and Problem Solving, May 19-20, 1980, NASA Langley Research Center.
- Organizer of the NASA Workshop on Automation and Future Missions in Space, Pajaro Dunes, California, June, 1980.

NASA AWARDS

- New Technology Award for "Analysis of Space Vehicle Structures Using the Transfer Function Concept," 1969.
- New Technology Award for "Finite Element Formulation for Linear Thermoviscoelastic Materials," 1970.
- New Technology Award for "Optimum Structural Design Based on Reliability Analysis," 1971.
- New Technology Award for "Analytical Procedure for Estimating Reliability of Randomly Encited Structures," 1971.
- U. S. Patent 3-568-874 for "Pressure Seal," 1971.
- New Technology Award for "Optimization of Structures on the Basis of Fracture Mechanics and Reliability Criteria," 1974.
- New Technology Award for "Analysis of Linear Viscoelastic Structures," 1979.
- Certificates of Recognition for the Creative Development of Technology for the years 1974 and 1979.

JOURNAL EDITOR

- Editor for "Robotics, Manipulators and Man-Machine Systems," International Journal of Mechanism and Machine Theory, since 1975.
- Associate Editor, ASME Journal for Computer Engineering, since 1981.

CONSULTING

- Consulting on Harbour Systems for Atom-Powered Vessels and Protective Structures, Hamburg, Germany, 1963.
- Consultant on Space Systems for EUROSAT, Geneva, Switzerland, 1980.
- Consultant on Robotic Systems, Logistic Technology International, since 1981.
- Consultant on Industrial Productivity, TRW, since 1982.

MISCELLANEOUS

- Conceived and developed speech controlled wheelchair/manipulator system for quadriplegic patients. The system was demonstrated to the U.S. Committee for the Employment of the Handicapped by invitation of the President of the United States and to the United States Congress in 1976.

INDUSTRIAL EMPLOYMENT

JET PROPULSION LABORATORY

Program Manager

Autonomous Systems and Space Mechanics, 1976 to present.

- Responsible for leading and managing the JPL research and development program for autonomous systems and space mechanics for the space program.
- JPL Representative to the NASA Large Space Systems Technology Program, 1977 to 1981.
  JPL Representtive to the NASA Intercenter Working Group for Automated Operations, 1979 to 1982.
- JPL Representative to the NASA Payload Services Working Group, 1979 to 1981.
- Chairman, JPL Planing Committee for Autonomous Systems Technology, 1981.

Technical Manager

Advanced Technical Studies Office, February 1971-November 1976.

- Responsible for managing and directing advanced systems studies for the formulation of mission plans and development requirements for the Space Transportation System and for remote automamated exploration and operation in space.

Program Manager

Lunar Exploration Office, National Aeronautics and Space Administration Headquarters, February 1970-1971.

- On JPL assignment to NASA Headquarters, responsible for Apollo Lunar Surface Exploration System definitions, management and coordination involving several NASA Centers and contractors.

JPL (cont'd)	Research Supervisor Structural Mechanics, November 1966-February 1970. Responsible for the definition, direction, coordination, and management of the research and advanced development activities of the Dynamical and Structural Systems Research Group as related to unmanned space programs.
GENERAL ELECTRIC	Manager Space Mechanics Systems, General Electric, Missile Space Division, Space Science Laboratory, one year. Responsible for the definition, direction, coordination, and management of structural and mechanics systems research related to reentry systems, composite materials, flutter, control problems, thermal stresses, economic trade-off, operations research, etc. Scientist General Electric, Missile Space Division, Space Science Laboratory, one year. Studied and contributed to the theories of systems under random environmental inputs.
McDONNELL AIRCRAFT CORPORATION	St. Louis, one year. Established theoretical procedure for multi-component system analysis adaptable to experimental inputs and verification. Conducted theoretical investigations on the correlations between interacting systems when subject to random environmental inputs.
PAUL WEIDLINGER	New York, 3 1/2 years. Completely organized, managed, and executed structural and control dynamics analysis of Bell Laboratory TSX-1 Antenna for the Telstar System. In responsible charge of design for protection of equipment and structures against shock and excessive accelerations. Executed theoretical research studies on ocean and ground submerged structures subject to shock and random input.
HEWITT ROBINS	New York, three years. Responsible for the analysis and design of complex structures and automated machine systems for material and industrial handling.
DICKSON & EVANS	Toronto, Canada, two years. Responsible for analysis and design of machines and structures for industrial handling.
H.W. HINZ, Architects	Hamburg, Germany, three years. Planned, organized, supervised, and administered the construction work and associated financial affairs for four office buildings.

Ralph L. Hollis

Dr. Hollis received the B.S. degree in physics from Kansas State University in 1964, and the M. S. degree in physics, also from Kansas State University in 1965. From 1965 until 1970 he was employed by the Autonetics Division of North American Aviation, where he was engaged in computer simulation of space flight vehicles. Beginning in 1970, he attended the University of C 'orado, receiving the Ph.D. degree in solid state physics in 1975. After a brief postdoctoral appointment at the University of Colorado, he was a National Science Foundation / Centre Nationale de Researche Scientifique Exchange Scientist at Universite de Pierre et Marie Curie, Paris for part of 1976-77. In 1978, he joined IBM and has worked at the IBM Thomas J. Watson Research Center in fields of magnetism, acoustics, and robotics. He has received two IBM Invention Achievement Awards and is manager of the Robot Technology group in the Automation Research department.

LESLIE PACK KAELBLING

Computer Scientist
Artificial Intelligence Center
Computer Science and Technology Division

SPECIALIZED PROFESSIONAL COMPETENCE

Planning systems; commonsense reasoning; programming languages, and compilers

REPRESENTATIVE RESEARCH ASSIGNMENTS AT SRI

Implementation of a graphic simulation of robot design, and implementation of a verifiable language for robot control;
Research on situated planning and learning.

OTHER PROFESSIONAL EXPERIENCE

Research Assistant, John McCarthy, Stanford University; research on commonsense reasoning and planning
Teaching Fellow, Stanford University; taught introductory computer science courses.
Research Assistant, STAR Laboratory, Stanford University; computer programming

ACADEMIC BACKGROUND

A.B., Logic and Philosophy of Formal Systems (1983), Stanford University
Masters Student in Computer Science Science (1983-1984), Stanford University
Ph.D. Student in Computer Science (1984 to present), Stanford University
Research Fellow, Center for the Study of Language and Information (1985 to present), Stanford University

PROFESSIONAL ASSOCIATIONS AND HONORS

Association for Computing Machinery
American Association for Artificial Intelligence
Institute of Electrical and Electronics Engineers

July 1985

CURRICULUM VITAE

YUTAKA KANAYAMA

Artificial Intelligence Laboratory
Department of Computer Science
Stanford University
Stanford, California 94305

Born: February 12, 1937

Citizenship: Japan

Visa Status:
J-1 September 11, 1984 - September 11, 1989
B-1, B-2 August 13, 1981 - August 13, 1985

Education:
B.S. in Electrical Engineering
University of Tokyo, 1960

M.S. in Electrical Engineering
University of Tokyo, 1962

Ph.D. in Electronics
University of Tokyo, 1965

Professional Experience

1965-1968:
Research Fellow
Computer Science Department
Central Research Laboratory
Hitachi, Ltd.
Kokubunji, Tokyo 184, Japan

Main activities:
During this period, I designed the microprogramming and cpu architecture of

a HITAC pilot computer and the design- automation system for a new computer system.

1968-1971:
Assistant Professor
Electrical Engineering Department
Hosei University
Koganei, Tokyo 184, Japan

Courses taught:
Computability Theory
Elementary Mathematics

Main activities:
During this period, I taught courses in elementary mathematics and introductory computer sciences. I also pursued research in the field of context-free language and automata theory.

1971-1977:
Associate Professor
Computer Science Department
University of Electro-Communications
Chofu, Tokyo 182, Japan

Courses taught:
Artificial Intelligence
Language and Automata Theory
Computability Theory
Fortran Programming
Assembly Language Programming

Main activities:
Since this was a newly created department, I participated in the establishment of a computer center and a corresponding curriculum plan. I conducted research programs for many graduate students and was the chairman of the graduate course. My research activities were in the theory of computation and robotics during this time.

1977-1984:
Professor
Institute of Information Sciences
University of Tsukuba
Sakura, Ibaraki 305 Japan

Courses taught:
Artificial Intgelligence I,II,III
Language and Automata Theory
Robotics

Main activities:
I was the coordinating head of an elementary course in information processing for all freshmen in the University. No other university in Japan provides this type of course. My duties as head were administrative in nature. I was also the head of the committee to select a computer system for education use. My research activities include construction of a family of self-contained robots "Yamabicos," and a novel formal system, K-system, for providing equivalence of programs. More details about the robot project are given elsewhere in this vitae.

1984-present:
Research Associate
Artificial Intelligence Laboratory
Department of Computer Science
Stanford University
Stanford, CA 94305

Main activities:
I am involved in the robotics research group directored by Professor Thomas O. Binford and am acting as a subleader of the mobile robot project. The SMOOTH DRIVER system is a part of the accomplishment here. I am also responsible on subfields of real time control of mobil robots, model of the world for mobile robots and planning algorithm for mobile robots.
I am taking a part of the ALV(Autonomous Land Vehicle) project sponsored by DARPA. The Stanford group is cooperating a research group at AI&DS who is working on the planning system of ALV. Architecture, vision interface, terrain representation and planning are again included in my interests.

1978-1979:
Visiting Professor
Northwestern University
Department of Electrical Engineering and Computer Science
Evanston, IL 60201

Courses taught:
Operating System Evaluation and Measurement
Computer Architecture
Seminar in Artificial Intelligence

Main activities:
I pursued research in robotics and mathematical theory of programs with my colleagues in Japan. I also saw many professors and researchers at the University and other organizations engaged in the same research areas to discuss the latest results.

Professional Activities

Robotics Society of Japan:
Chief of the Editorial Board, 1983;
Director, 1983;
Member of Editorial Board, 1983-84

Main activities:
I am one of the founding members of this society. I had worked with Dr. Nakano of the Mechanical Engineering Laboratory of MITI to organize the society before it was was founded in January 1983. I am now in charge of establishing an international edition of an international edition of the Journal of the Robotics Society of Japan.

International Symposium on Industrial Robots:
Member of the Organizing Committee for the 15th ISIR in 1985.

International Conference on Advanced Robotics:
Member of the Organizing Committee for the 1985 ICAR.

Association for Computing Machinery: Member.

IEEE Computer Society: Member.

Japan Industrial Robot Association:
Member of Special Interest Group on Standardization of Robot Languages.

Institute of Electronics and Communication Engineers of Japan: Referee.

Information Processing Society of Japan: Member.

Japan Atomic Energy Research Institute:
Advisory Committee on Dismantling Atomic Power Plant, and research consultant on their project of disassembling an old atomic power plant.

Technological Research Association of Medical and Welfare Apparatus:
Advisory Committe on Development of Patient Care Mobile Robot for the Handicapped. I participate in designing patient care mobile robots.

Japan Micromouse Association:
Consultant on defining rules and refereeing for the Japan Micromouse Contest held every year and on preparing the World Micromouse Contest.

Current Interests

Mobile Robots:
Multiprocessor Architecture for Real Time Robot Control, Real Time Operating System, Trajectory Generation, Representing 2-D Environment, Understanding Environment using Sonic Range Finding, Laser Range Finding and Vision System, Planning, Concurrent Process Support.

Robot Manipulator: Standard Language

Computer Vision:
3D Recognition by Active Control of Viewpoints, Dynamic Scene Analysis on Mobile Robots, Line Finding by using Area Generation.

Artificial Intelligence:
Representation of Robot Intelligence, Production System/Prolog.

Theory of Programs: Equivalence, Correctness, Complexity.

David Miller
Curriculum Vita

October 14, 1985

David Miller
411 Clay S.W.
Blacksburg, Va 24060
(703) 953-0013

Department of Computer Science
Virginia Polytechnic Instititute
Blacksburg, Va 24061
(703) 961-6075

EDUCATION:

1981: B.A. With Honors in Astronomy, Wesleyan University, Middletown, Ct.

1985: Ph.D. Computer Science, Yale University, New Haven, Ct.

POSITIONS HELD:

Fall 85–present: Assistant Professor of Computer Science, Viriginia Polytechnic Institute and State University, Blacksburg, Virginia

Fall 81–Spring 85: Research Assistant, Yale University, Department of Computer Science, New Haven, Connecticut

Fall 84: Visiting Instructor, Wesleyan University, Department of Mathematics, Middletown, Connecticut

Spring 82–Spring 84: Teaching Assistant, Yale University, Department of Computer Science, New Haven, Connecticut

CAREER HIGHLIGHTS

Name:	Edward M. Oblow		
Education:	C.C.N.Y., New York	B.S. Chem. Eng.	1965
	Columbia Univ., N.Y.	M.S. Nucl. Eng.	1967
	Columbia Univ., N.Y.	Eng. Sci. D. Appl. Phys.	1970
Work:	Brookhaven National Lab. Research Assoc.		1964
	X-ray experiments		
	Columbia University Research Assoc.		1970-71
	Neutron transport and cross section theory		
	Oak Ridge National Lab. Senior Research Staff		1972 - present
	Radiation transport and cross section theory and experiments, sensitivity theory, fusion reactor shielding, plasma physics theory and experiments, fast reactor safety and thermal hydraulics, energy-economy modeling		
Publications:	Approximately 100 reports and journal publications with major efforts in diverse aspects of sensitivity theory, fusion and fission reactor shielding (both experimental and theoretical), and neutron cross section and transport theory		
Meetings:	Represented U.S. in three international specialists meetings in reactor shielding and sensitivity theory in France, Austria and Russia		

College of Engineering
RESEARCH RESUME

KARL N. REID, JR.

Professor and Head
School of Mechanical and Aerospace Engineering
Oklahoma State University

ACADEMIC BACKGROUND

Sc.D., Mecnanical Engineering, Massachusetts Institute of Technology, 1964
M.S., Mechanical Engineering, Oklahoma State University, 1958
B.S., Mechanical Engineering, Oklahoma State University, 1956

MAJOR AREAS OF INTEREST

Systems Analysis; Automatic Control
Fluid Control Systems
Instrumentation; Computer-Aided Design

PROFESSIONAL EXPERIENCE

Head, School of Mechanical and Aerospace Engineering, Oklahoma State University, 1972-present; Professor, 1970-present; Associate Professor, 1967-1970; Assistant Professor, 1964-1967.

Chairman, CAD/CAM and Robotics Thrust, College of Engineering, Oklahoma State University, 1981-present.

Director, Center for Systems Science, College of Engineering, Oklahoma State University, 1968-1972.

Instructor, Department of Mecnanical Engineering, Massachusetts Institute of Technology, 1960-1964; Coordinator of Engineering Projects Laboratory, 1961-1963; Teaching Assistant, 1959-1960; Research Assistant, 1958-1959.

Engineer, Office of the Post Engineer, U.S. Army, Ft. Belvoir, Virginia, 1958.

Engineer, Research Department, Vickers Inc., Detroit, Michigan, Summer 1957.

Consultant to various firms including: Westinghouse Electric Corp., Gulf Insurance Co., Scovill Manuf. Co., Liberty Mutual Ins. Co., Economics Laboratory, Air Products & Chemicals Inc., Stihl-American Co., National Institutes of Health, O. M. Scotts, Inc., Exxon Corp.

Educational Consultant to the National Science Foundation and U.S. Agency for International Development; Organizer and lecturer at Summer Institutes on Fluidics and Control Engineering, Coimbatore, India, Summers 1969 and 1970.

Co-organizer and lecturer for two-week courses on Fluid Power Control at M.I.T., 1959, 1962, and U.C.L.A., 1961; Co-organizer and lecturer for two-week course on instrumentation for Measurement and Control at M.I.T., 1963; Lecturer, two-week workshops on Fluid Control Systems at M.I.T., 1965, and Penn. State, 1966; Director and Lecturer, two-week workshop on Fluid Control Systems at OSU, 1968.

RESEARCH EXPERIENCE

Principal Investigator, Oak Ridge National Laboratories, research needs assessment in intelligent machines for unstructured environments, 1983-1984

Principal Investigator, National Science Foundation grant to the A.S.M.E., national research needs assessment in design methods and computer graphics, machine dynamics, tribology, dynamic systems and control, 1982-1984.

Project Director, Moog Servocontrols Inc., dynamic behavior of electrohydraulic servovalves, 1980-present.

Project Co-Director, Corps of Engineers, hydraulic transients in pumping systems, 1976-1977.

Project Director, U.S. Army Harry Diamond Labs, hydraulic vortex resistance devices, 1974-1976.

Project Director, General Electric Co., vehicle propulsion systems employing hydromechanical transmissions, 1973-1975.

Project Director, U.S. Army Harry Diamond Labs, fluidic temperature control in vehicle propulsion systems, 1973.

Project Director, National Science Foundation and State of Oklahoma sponsored Center for Systems Science, 1969-1972.

Project Director, Boeing Co., synthesis of human operator mathematical models for multiple-axis tasks, 1969-1970.

Project Director, U.S. Army Tank Automotive Center, the application of fluidics in military vehicle propulsion systems, 1967-1972.

Project Director, Allis Chalmers, pulsating flow hydraulic systems, 1966-1969.

Project Director, U.S. Army Harry Diamond Labs, fluid flow phenomena in fluidic devices, 1964-1969.

PROFESSIONAL AFFILIATIONS

National Science Foundation - Member, Advisory Sub-Committee, Mechanical Engineering and Applied Mechanics Division, 1981-1984.

Massachusetts Institute of Technology Corporation - Member, Visiting Committee for Mechanical Engineering (1979-present).

American Society of Mechanical Engineers - Vice President, Board on Communications (1984-1986); Consulting Editor (1980-1982), Sr. Technical Editor (1976-1980), Editor (1974-1976), Assoc. Editor (1971-1973), ASME Transactions/Journal of Dynamic Systems, Measurement and Control; Chairman (1980-1981), Vice Chairman (1978-1980), Member (1976-1981), Executive Committee, Dynamic Systems & Control Division; Chairman, Fluidics Committee (1968-1970); Member, Applied Mechanics Reviews Advisory Committee (1979-1982); Member, Publications Committee, Policy Board Communications (1979-1981); Member, Board on Communications (1981-1984); Chairman, ASME Transactions Board of Editors (1979-1982); Member Executive Committee, Central Oklahoma Section (1972-present).

American Institute for Aeronautics and Astronautics (1979-present).

American Society of Engineering Education (1964-present); Chairman, Publication Policy Board, ME Division (1981-1984); Associate Editor of Mechanical Engineering News (1968-1978); Member, Sr. Research Award Committee (1976-1982).

PROFESSIONAL AFFILIATIONS

International Federation for Automatic Control - U.S. Representative to International Technical Committee on Components (1970-1974 and 1979-1984).
American Automatic Control Council - ASME Director (1983-1984); Chairman, Components Committee, (1972-1981); Chairman, Instrumentation Committee (1981-1983).
Oklahoma Society of Professional Engineers (1974-present).
National Society of Professional Engineers (1974-present).
Fluid Power Society (1962-present).

HONORS AND AWARDS

Elected ASME Fellow, 1983
ASME Centennial Medallion, 1980
Outstanding Teacher in College of Engineering, 1967
Western Electric Outstanding Teacher Award, 1972
Oklahoma Society of Professional Engineers:
Outstanding Engineering Achievement Award for a "Center for Systems Science at Oklahoma State University," 1971; Outstanding Engineering Achievement Award for the "Development of a Fluidically Controlled Lung Ventilator," 1973; Wonder of Engineering Award for the "Development of an External Heart Massage Device," 1973
Executive of the Year, Stillwater Chapter, National Secretaries Association, 1980
American Men of Science
Pi Tau Sigma, Sigma Tau, Phi Kappa Phi, Sigma Xi, Blue Key, Omicron Delta Kappa

BIOGRAPHICAL SKETCH of Bill D. Richard
(compiled 9-25-85)

WORK ADDRESS: Sandia National Laboratories
Exploratory Development Division - 5268
P. O. Box 5800
Albuquerque, NM 87185
(505) 844-8414

HOME ADDRESS: 2709 Compa Ct. NE
Albuquerque, NM 87112
(505) 293-4017

EDUCATION: Bachelor of Science - Summa Cum Laude, Mathematics/Physics, 1974, Eastern New Mexico University

Master of Science, Mathematics/Computer Science, 1976, Texas Tech University

Additional graduate level courses in Mathematics, Computer Science and Electrical Engineering completed through Sandia National Laboratories

Undergraduate awards received in Mathematics, Physics, and Military Science

Member of Phi Kappa Phi and Sigma Pi Sigma

PROFESSIONAL EXPERIENCE: 4-85 to present - Sandia National Laboratories, Member of Technical Staff, Systems Analyst/Computer Scientist, Software Project Leader for the Mobile Surveillance/Assessment Platform (an autonomous land vehicle) Project.

12-80 to 3-85 - Sandia National Laboratories, Member of Technical Staff, Computer Scientist in the area of Computer-Aided Design of Integrated Circuits. Projects included IC Layout, Artwork Generation and Design Verification.

6-76 to 11-80 - Sandia National Laboratories, Member of Laboratory Staff, Systems Analyst specializing in scientific systems development involving statistical and digital signal processing applications.

PERSONAL DATA: Date of Birth - May 28, 1952
Wife - Charla Annette Richard
Children - William Nathaniel Richard (4-3-81)
- Jonathan David Richard (4-16-84)

DR. ROBERT B. TILOVE

Robert B. Tilove received his Ph.D. in Electrical Engineering from the University of Rochester in 1981. He is a senior staff research scientist in the Computer Science Department of General Motors Research Laboratories in charge of research in machine perception and robotics. Prior to joining GM in 1981, Dr. Tilove was a research assistant with the Production Automation Project at the University of Rochester and chief analyst of the PADL-2 solid modeling project. His research interests include geometric modeling, CAD/CAM, computer vision, and autonomous mobility.

September 1985

STANLEY J. ROSENSCHEIN

Director
Artificial Intelligence Center
Computer Science and Technology Division

SPECIALIZED PROFESSIONAL COMPETENCE

Artificial intelligence, computational linguistics, programming languages

REPRESENTATIVE RESEARCH AT SRI (since 1980)

Project leader, ONR contract: "Distributed Artificial Intelligence"
Research on natural language semantics and formal models of planning

OTHER PROFESSIONAL EXPERIENCE

Associated Computer Scientist, RAND Corporation, Santa Monica, CA.
Lecturer, Technion--Israel Institute of Technology, Haifa, Israel
Associate Research Scientist, Courant Institute, New York University

ACADEMIC BACKGROUND

A.B. (1971), Columbia University
M.S.E. (1973), Electrical Engineering, University of Pennsylvania
Ph.D. (1975), Computer Science, University of Pennsylvania

REPRESENTATIVE PUBLICATIONS

"Plan Synthesis: A Logical Perspective," *Proceedings of International Joint Conference on Artificial Intelligence*, University of British Columbia, Vancouver, British Columbia (1981)

"Abstract Theories of Discourse and the Formal Specification of Programs that Converse," *Elements of Discourse Understanding*, A. K. Joshi, B. Webber, and I. Sag (eds.), Cambridge University Press (1981)

"The Production System: Architecture and Abstraction," in D. A. Waterman and F. Hayes-Roth (eds.), *Pattern-Directed Inference Systems*. New York: Academic Press (1978)

Coauthor "Schenker's Theory of Tonal Music--Its Explication through Computational Processes," *International Journal of Man-Machine Studies*, 10, pp. 121-138 (1978)

Coauthor "Making Computational Sense of Montague's Intensional Logic," *Artificial Intelligence*, 9 pp. 287-306 (1977)

Coauthor "Some Problems of Inferencing: Relations of Inferencing to Decomposition of Predicates," in *Statistical Methods in Linguistics*. Skriptor, Sweden (1976)

Coauthor "A LISP-Based System for the Study of Schenkerian Analysis, *Computers and the Humanities*, 10, pp. 21-32 (1976)

"How Does a System Know When to Stop Inferencing," *American Journal of Computational Linguistics*, Microfiche 36, (Winter 1975)

PROFESSIONAL ASSOCIATIONS AND HONORS

Executive Committee, Association for Computational Linguistics
Reviewer, International Joint Conference on Artificial Intelligence, JACM, Computing Machinery
Association for Computing Machinery
Cognitive Science Society
American Association for Artificial Intelligence

C. R. Weisbin

119 Newhaven Road
Oak Ridge, TN 37830
Date of Birth - 1-4-44
Citizenship - USA
Sex - Male

Employee Number - 012772
Company Service Date - 9-10-73
Clearance - Current Q
Married - two children
Social Security # 080-34-9483
Business - (615)-574-6186
Home - (615)-482-4886

ACADEMIC EDUCATION

- Foreign Languages - Read, French - German

1969 - Doctorate Degree, Columbia University, New York, New York
Major - Nuclear Engineering and Nuclear Reactor Engineering
Minor - Applied Physics
Thesis Title - "A New Moments Solution to the Neutron Transport Equation"

1965 - Masters Degree, Columbia University, New York, New York
Major - Nuclear Engineering and Nuclear Reactor Engineering
Minor - Applied Physics

1964 - Bachelors Degree, Polytechnic Institute of Brooklyn, New York
Major - Chemical Engineering and Nuclear Chemistry
Minor - Chemistry

ADDITIONAL ACADEMIC EDUCATION

8/1981 - 3.6 continuing education units, The George Washington University, School of Engineering and Applied Science - Applications of Reliability and Risk Analysis with Emphasis on Nuclear Power Plants

ACADEMIC POSITIONS

9/1984 to - present: Associate Professor of Computer Science, University of Tennessee
Class: Artificial Intelligence

10/1985 to- present: Member, Editorial Board of "Expert Magazine: Intelligent Systems and Their Application."

1/1984 to - present: Member, Editorial Board of Nuclear Science & Engineering

PROFESSIONAL AND ACADEMIC HONORS

1965 - Sigma Chi
1964 - B.S. Nuclear Engineering Cum Laude
1964 - Tau Beta Pi
1962-64 - AEC Fellow, Dean's List

TECHNICAL SPECIALTIES

Perform Work - Artificial Intelligence
Robotics
Process Control Systems
Energy Supply, Demand and Forecasting
Reactor Physics

Career Interests - Artificial Intelligence
Robotics
Process Control Systems
Reactor Physics
Energy Technology Assessment

PROFESSIONAL EXPERIENCE

11/82 - Present	Director of the Center for Engineering Systems Advanced Research (CESAR)
12/80 - Present	Section Head, Simulation and Evaluation of Energy Systems, Administer the activities of two groups within the Section
10/77 - 12/80	Group Leader, Reactor Methods and Data Development, Engineering Physics and Mathematics Division, ORNL
9/10/73-10/77	Project Leader, Engineering Physics and Mathematics Division, ORNL
4/70 - 9/73	Research Staff Member, Nuclear Data Research Group, Los Alamos National Laboratory
5/69 - 4/70	Research Associate, Columbia University - Post Doctoral Research for Professor H. Goldstein

PROFESSIONAL SOCIETIES MEMBERSHIP

Technical Committee - Pattern Analysis and Machine Intelligence
Institute of Electrical and Electronic Engineers
American Society of Mechanical Engineers
American Nuclear Society
Robotics International/SME
American Association for the Advancement of Science
American Defense Preparedness Association
Tau Beta Pi
Sigma Xi

SPECIAL LABORATORY/DOE/MARTIN MARIETTA ASSIGNMENTS

1985-Present	Martin Marietta Corporate Steering Committee on Artificial Intelligence
1985-Present	Martin Marietta Energy Systems, Inc., Energy Technologies Committee
5/83 - Present	Director, Center for Engineering Systems Advanced Research
2/82 - 7/83	Director, Carbon Dioxide Information Center
1979 - 81	Leader, ORNL Model Validation Group
1975 - 81	EPRI Coordinator for Nuclear Data Requirements and Applications
1980	UCND Management Course

OTHER SPECIAL ASSIGNMENTS

1985	Chairman and Organizer, 1985 DOE/ORNL National Workshop on "Planning and Sensing for Autonomous Navigation."
1985	Program Chairman, Second International Conference on Artificial Intelligence Applications, IEEE
1980-1982	Program Committee Member, 1982 Reactor Physics and Thermal Hydraulics Topical Meeting, Kiamesha Lake
1979-1980	Technical Program Committee for the Reactor Physics, Idaho Topical Meeting, September 14-18, 1980
1979-1981	ASTM E10.05 Subcommittee (Standardization of Dosimetry as Part of E10 Committee on Nuclear Technology Application)
1977-1983	Cross Section Evaluation Working Group, Data Testing and Applications Committee Chairman
1972-1983	Committee on Computer Code Coordination
1972-1984	ANS 19: Physics of Reactor Design Standards Committee
1971-Present	American Nuclear Society Shielding, Mathematics and Computation, and Reactor Physics Divisions
1971-1982	Reactor Physics Division Technical Program Committee
1971-1983	Code Evaluation Working Group

OTHER SPECIAL ASSIGNMENTS (continued)

1971-1983	Cross Section Evaluation Working Group (Codes and Formats, Data Testing, Fission Products, and Shielding)
1977-1979	Reactor Physics Division Program Committee Chairman
1965-1979	Reactor Physics Division Program Committee Chairman

CURRENT SUPERVISORY RESPONSIBILITY

Monthly - 22
Weekly - 5

DISTRIBUTION LIST

DISTRIBUTION LIST

ORNL/TM-9923
CESAR-86/01

INTERNAL DISTRIBUTION

1- 5. Abbott, L. S.
6. Anderson, T. D.
7. Babcock, S. M.
8. Barhen, J.
9. de Saussure, G.
10. Eads, B. G.
11. Einstein, J. R.
12. Ford, W. E.
13. Fulkerson, B.
14. Hall, M. C. G.
15. Hamel, W. R.
16-20. Howe, E. S.
21. Jorgensen, C. C.
22. Lucius, J. L.
23-27. Maienschein, F. C.
28. Mynatt, F. R.
29. Oblow, E. M.
30. Pin, F. G.
31. Roberto, J. B.
32-36. Weisbin, C. R.
37. Zucker, A.
38. Dickson, P. W., Jr. (Consultant)
39. Golub, G. H. (Consultant)
40. Haralick, R. (Consultant)
41. Steiner, D. (Consultant)
42-43. Central Research Library
44. ORNL Y-12 Technical Library
Document Reference Section
45-46. Laboratory Records
47. ORNL Patent Office
48. Laboratory Records - RC
49-53. EPMD Reports Office

EXTERNAL DISTRIBUTION

54. R. E. Balzhiser, Vice President, Research & Development, Electric Power Research Institute, P.O. Box 10412, Palo Alto, CA 94303.

55. Stephen D. Ban, Vice President Research, Gas Research Institute, 8600 West Fryn Mawr, Chicago, IL 60631.

56. Dr. Antal K. Bejczy, Robotics and Teleoperator Group, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109.

57. A. Bement, Vice President, Technical Resources - TRW, 1900 Richmond Road, Cleveland, OH 41124.

58. Prof. Thomas O. Binford, Artificial Intelligence Laboratory, Margaret Jackson Hall, Stanford University, Palo Alto, CA 94305.

59. Ted Blank, Editor-in-Chief, Smart Machines, P.O. Box 459, Sharon, Massachusetts 02067.

60. Dr. Wayne Book, Department of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

61. Dr. Judy Bostock, Office of Management & Budget, Executive Office Building, 17th and Pennsylvania Avenues, Washington, DC 20503.

62. Dr. John F. Cassidy, General Electric Corporate R&D, Bldg. 5 - Room 249, Control Technology Branch, Schenectady, NY 12345.

63. Raymond GA Cate, Editor, Robotics Age, Strand Building, 174 Concord Street, Peterborough, New Hampshire 03458.

64. B. Chandrasekaran, Laboratory for Artificial Intelligence Research, Department of Computer & Information Science, 2036 Neil Avenue Mall, Columbus, OH 43210-0923.

65. Peter Cheeseman, Robotic Department, SRI International, 333 Ravenswood Road, Menlo Park, CA 94025.

66. Don Ciffone, National Aeronautics and Space Administration, Ames Research Center, MS 244-7, Moffett Field, CA 94035.

67. Dr. J. S. Coleman, Division of Engineering, Mathematical, and Geosciences, Office of Basic Energy Sciences, ER-15, U.S. Department of Energy-Germantown, Washington, DC 20545.

68. Professor Jim Crowley, Carnegie-Mellon University, Robotics Institute, Pittsburgh, PA 15213.

69. J. B. Cruz, Jr., Professor, Coordinated Science Laboratory, University of Illinois, 1101 W. Springfield Avenue, Urbana, IL 61801.

70. Dr. Martin J. Dudziak, Martin Marietta Aerospace, Baltimore Division, 103 Chesapeake Park Plaza, Baltimore, MD 21220.

71. Dr. Alberto Elfes, The Robotics Institute, Carnegie Mellon University, Schenley Park, Pittsburgh, PA 15213.

72. Robert Engelmore, Editor-in-Chief AI Magazine, Teknowledge, Inc., 525 University Avenue, Palo Alto, CA 94301

73. Capt. Kristin A. Farry, Flight Dynamics Laboratory, Bldg. 146, AFWAL/FIGC/58683, Wright-Patterson Air Force Base, Ohio 45433.

74. R. J. Goldstein, Professor and Head, Department of Mechanical Engineering, University of Minnesota, 111 Church Street, SE, Minneapolis, MN 55455.

75. Professor R. C. Gonzalez, Department of Electrical Engineering, University of Tennessee, Knoxville, TN 37916.

76. Dr. Charles A. Hall, Martin Marietta Aerospace, Baltimore Division 103 Chesapeake Park Plaza, Baltimore, MD 21220

77. Scott Y. Harmon, Code S442, Naval Ocean Systems Center, San Diego, CA 92152.

78. Ewald Heer, Heer Associates, Inc., 5329 Crown Avenue, LaCanada, CA 91011.

79. Dr. David B. Hertz, Director, Intelligent Computer Systems Research Institute, University of Miami, Miami, FL.

80. Steven W. Holland, General Motors Research Laboratories, Computer Science Department, GM Technical Center, Warren, MI 48090-9057.

81. Frank Holley, Human Engineering Laboratory, ATTN: ANXHE-FH (Holley), Aberdeen Proving Grounds, Maryland 21005.

82. Dr. Ralph L. Hollis, Research Staff Member, Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598.

83. Ramesh Jain, Associate Professor, The University of Michigan, Department of Electrical Engineering and Computer Science, Division of Computer Science and Engineering, Ann Arbor, MI 48109-1109.

84. M. J. Jefferies, Manager, Research & Development, Engineering Physics Laboratories, General Electric Company, P.O. Box 8, Schenectady, NY 12301.

85. Dr. Leslie Kaelbling, SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025.

86. Avi Kak, Department of Electrical Engineering, Purdue University, Lafayette, IN 47907.

87. Yutaka Kanayama, Stanford University, Cedar Hall, Artificial Intelligence Laboratory, Stanford, CA 94305.

88. David Keirsey, Hughes AI Center, 23901 Calabasas Road, Calabasas, CA 91302.

89. J. Kestin, Professor, Engineering Division, Brown University, Providence, RI 02912.

90. J. D. Lowrance, SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025.

91. R. E. Lyon, Jr., Vice President, Exxon Research and Engineering Company, P.O. Box 101, Florham, Park, NJ 07932.

92. Dr. Oscar P. Manley, Division of Engineering, Mathematical, and Geosciences, Office of Basic Energy Sciences, U.S. Department of Energy - Germantown, Washington, DC 20545.

93. Stephen H. Maslen, Associate Director, Martin Marietta Laboratories, 1450 South Rolling Road, Baltimore, MD 21227.

94. Harvey B. Meieran, H. B. Meieran Associates, 458 South Dallas Avenue, Pittsburgh, PA 15208.

95. Alexander Meystel, Drexel University, ECE, Philadelphia, PA 19104.

96. David Miller, Department of Computer Science, 562 McBryde Hall, Virginia Polytechnic Institute, Blacksburg, VA 24061.

97. Scott D. Myers, Staff Engineer, Systems Analysis and Simulation, Martin Marietta Aerospace, Baltimore Division, 103 Chesapeake Park Plaza, Baltimore, MD 21220.

98. Professor David E. Orin, Department of Electrical Engineering, The Ohio State University, 2015 Neil Avenue, Columbus, OH 43210.

99. Dr. John F. Palmer, Chairman, NCUBE Corporation, 915 E. La Vieve Land, Tempe, AZ 85284.

100. Alexandre Parodi, Artificial Intelligence Center, FMC Corporation, Central Engineering Laboratories, 1185 Coleman Avenue, Box 580, Santa Clara, CA 95052.

101. Dr. Alex Penland, SRI International, Room EK243, 333 Ravenswood Avenue, Menlo Park, CA 94025.

102. Prof. Harry E. Pople, Decision Systems Laboratory, University of Pittsburgh, School of Medicine, Pittsburgh, PA 15261.

103. Dr. Mike Rabins, Mechanical Engineering Department, Wayne State University, Detroit, MI 48202.

104. Carlos Ramirez, Staff Engineer, Robotic Processes, Martin Marietta Aerospace, Advanced Manufacturing Technology, Michoud Division, P.O. Box 29304, M/S 3691, New Orleans, Louisiana 70189.

105. Professor Karl N. Reid, Head, School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK 74074.

106. Howard Resnikoff, Vice President and Director of Research, Thinking Machine Corporation, 245 First Street, Cambridge, MA 02142-1214.

107. Professor W. C. Reynolds, Department of Mechanical Engineering, Stanford University, Stanford, CA 94305.

108. Dr. Bill Richard, Exploratory Development, Division 5628, Sandia National Laboratories, Albuquerque, NM 87185.

109. Dean H. W. Richardson, Engineering Department, Texas A&M University, College Station, TX 77843.

110. John Roach, Department of Computer Science, 562 McBryde Hall, Virginia Polytechnic Institute, Blacksburg, VA 24061.

111. Dr. Guillermo Rodriguez, Machine Intelligence Systems, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109.

112. Stan Rosenschein, Stanford Research Institute, 333 Ravenswood Avenue, Menlo Park, CA 94025.

113. Professor George Saridis, Electrical, Computer, and Systems Engineering Department, Rensselaer Polytechnic Institute, 15th Street, Troy, NY 12180.

114. Roger T. Schappell, Aerospace Division, Martin Marietta Aerospace, P.O. Box 179, Denver, CO 80201

115. R. Schmitz, Dean of Engineering, College of Engineering, University of Notre Dame, Notre Dame, IN 46556.

116. Jacob T. Schwartz, Courant Institute of Mathematical Science, New York University, 251 Mercer Street, New York, NY 10012.

117. G. Shafer, Department of Mathematics, University of Kansas, Lawrence, KS 66045.

118. Director, U.S. Army Human Engineering Laboratory, ATTN: AMXHE-CS (Shoemaker), Aberdeen Proving Grounds, Maryland 21005-5001.

119. R. Sobek, Laboratoire d'Automatique, et d'Analyse des Systemes, du C.N.R.S. 7, avenue du Colonel-Roche, F-31077 TOULOUSE CEDEX, FRANCE

120. Andy Spiessbart, Martin Marietta Aerospace, P.O. Box 179, Mail Stop 0570, Denver, CO 80201.

121. Robert N. Stouffer, Editor, Robotics Today, One SME Drive, P.O. Box 930, Dearborn, MI 48121.

122. Professor Delbert Tesar, Carol Cockrell Curran Chair in Engineering, The University of Texas at Austin, Room: ETC 4.146C, Austin, TX 78712.

123. Perry W. Thorndyke, Director, Artificial Intelligence Center, FMC Corporation, Central Engineering Laboratories, 1185 Coleman Avenue, Box 580, Santa Clara, CA 95052.

124. Chang-Lin Tien, Professor, Mechanical Engineering Department, University of California, Berkeley, CA 94720.

125. Robert B. Tilove, General Motors Research Laboratories, Computer Science Department, GM Technical Center, Warren, MI 48090-9057.

126. Harlan L. Watson, Committee on Science Technology, U.S. House of Representatives, Washington, DC 20515.

127. Dr. John Weisz, U. S. Army Human Engineering Laboratory, Attn: DRX HE-D, Aberdeen Proving Grounds, MD 21009.

128. Chester A. Winsor, Martin Marietta Aerospace, 6801 Rockledge Drive, Bethesda, MD 20817.

129. Pat H. Winston, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, MA 02139.

130. L. Zadeh, Computer Science Department, University of California, Berkeley, CA 94720.

131. Assistant Manager, Energy Research and Development, DOE/ORO, Oak Ridge, TN 37831.

132-158. Technical Information Center, P. O. Box 62, Oak Ridge, TN 37831.

U.S. GOVERNMENT PRINTING OFFICE: 1986-631-056/20217