NASA Conference Publication 2180

NASA-CP-2180-VOL-2
19810021364

Automated Decision Making and Problem Solving

Volume II - Conference Presentations

Proceedings of a conference held at
NASA Langley Research Center
Hampton, Virginia
May 19-21, 1980

NASA

NASA Conference Publication 2180

Automated Decision Making and Problem Solving

Volume II - Conference Presentations

Edited by
Ewald Heer
University of Southern California
Los Angeles

Proceedings of a conference held at
NASA Langley Research Center
Hampton, Virginia
May 19-21, 1980

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1981

PREFACE

On May 19-21, 1980, NASA Langley Research Center hosted a Conference on Automated Decision Making and Problem Solving. The purpose of the conference was to explore related topics in artificial intelligence, operations research, and control theory and, in particular, to assess existing techniques, determine trends of development, and identify potential for application in automation technology programs at NASA. The first two days consisted of formal presentations by experts in the three disciplines. The third day was a workshop in which the invited speakers and NASA personnel discussed current technology in automation and how NASA can and should interface with the academic community to advance this technology.

The conference proceedings are published in two volumes. Volume I gives a readable and coherent overview of the subject area of automated decision making and problem solving. This required interpretation, synthesizing, and summarizing, and in some cases expansion of the material presented at the conference. Volume II contains the vugraphs with various annotations extracted from videotape records and also written papers submitted by several authors. In addition, a summary of the issues discussed on the third day has been published separately in NASA Technical Memorandum 81846.

CONTENTS

NASA CROSSCUT STUDIES AND APPLICATIONS

William B. Gevarter

NASA Headquarters

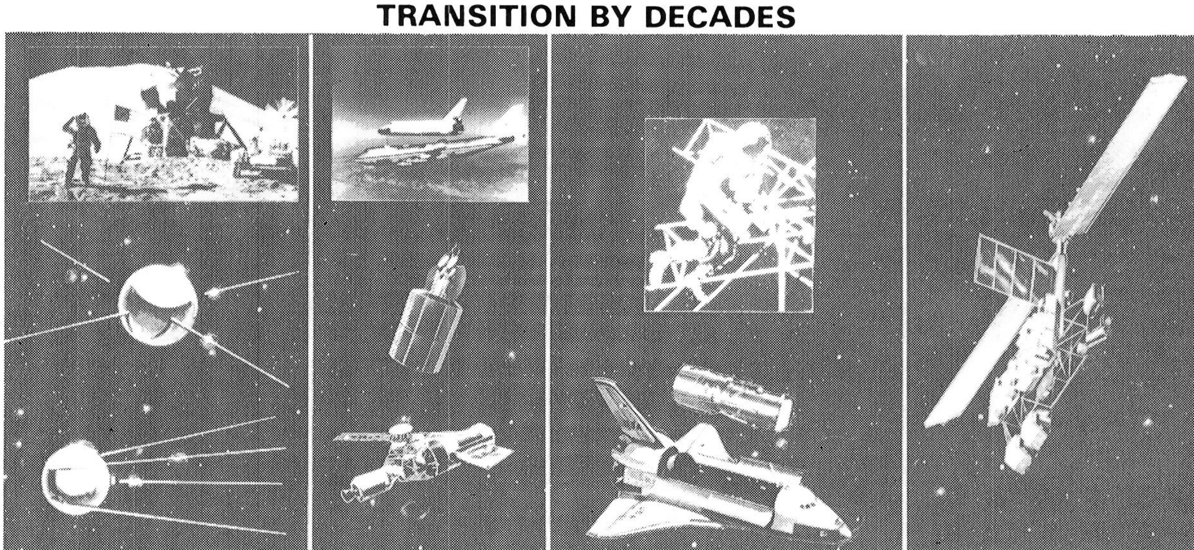
This is an indication of the history of NASA's activities. Landing on the Moon indicated to us that we can do almost anything if we apply enough resources to it--we established technological feasibility in the 1960's.

In the 1970's, we demonstrated that we could do certain things which had utilitarian value; e.g. Landsat, communications programs, platforms in space, lunar and asteroid mining, etc.

The problem is, however, we can not afford all of it. So we have to establish economic feasibility which will be the prime concern for the 1980's space program. This concern will be with us also into the 1990's. We have to demonstrate that space utilization is not only beneficial but also affordable.

SPACE PROGRAM EVOLUTION

TRANSITION BY DECADES



1960's	1970's	1980's	1990's AND BEYOND
TECHNOLOGICAL FEASIBILITY	UTILITY DEMONSTRATION	ECONOMIC FEASIBILITY	AFFORDABLE, BENEFICIAL MISSIONS

Within the automation structure, we have various areas. The things which are listed here fall within generic technologies and application classes. This is, of course, not the only possible subdivision of what we call automation.

AUTOMATION STRUCTURE

GENERIC TECHNOLOGY

- DECISION MAKING AND LOGIC
- KNOWLEDGE REPRESENTATION
- PERCEPTION
- LEARNING
- HUMAN/MACHINE INTERACTION
- MANIPULATION
- MOBILITY

APPLICATION CLASSES

- OPERATIONS SUPPORT AND PLANNING
- TELEFACTOR SYSTEMS
- AUTONOMOUS SPACECRAFT OPERATION
- INFORMATION EXTRACTION

Some of the main points leading up to the automation program are shown here.

Mission costs, e.g., ground operations of long duration missions began to get out of hand.

The "Sagan Report" concluded that NASA appears to be behind the computer science cutting edge technology between 5-15 years on the average. This may be arguable at least in some areas. The "average" allows one to be ahead in some areas while being behind in others. In very narrow specific areas, NASA seems to be in the forefront if not considerably ahead.

The 1979 Innovators Meeting at Woods Hole, attended by Dr. Frosch, came up with the idea of self-replicating telefactors. If such an idea could be translated into practice, the impact on the affordability of space systems could be tremendous. Machines (robots) would be able to replicate and multiply exponentially. The only problem of course is how to build the first robot to self-replicate. A one-week Goal-Setting Workshop, and a ten-week Feasibility Workshop this summer will address these questions.

In response to these developments, we redirected OAST technology activities into the automation of selected ground operations and focused the technology on machine intelligence while the workshop planning studies shown here are ongoing and NASA automation activities across the agency are being reviewed.

Automated decision-making and problem solving is a major area in the OAST program.

AUTOMATION PROGRAM HISTORY

BACKGROUND

HIGH MISSION SUPPORT COSTS

"SAGAN REPORT"

'79 INNOVATORS MEETING

OAST RESPONSES

REDIRECTED OAST TECHNOLOGY ACTIVITIES

- AUTOMATION OF SELECTED GROUND ACTIVITIES
- TECHNOLOGY FOCUS ON MACHINE INTELLIGENCE

SUMMER WORKSHOPS

1. LONG-RANGE GOALS WORKSHOP: . . . JUNE 15-22
2. MISSION/TECHNOLOGY FEASIBILITY WORKSHOP: . . . . JUNE 23-AUG. 29

CROSS-CUT REVIEW OF NASA AUTOMATION ACTIVITIES

Here we discuss briefly the Automation Program History, including background and corresponding OAST responses. The NASA program office needs are briefly identified in the NASA crosscut activities.

One of the problems is that we tend to do something about things which belong in neat little boxes. There is no box for "efficiency," so we do nothing directly for efficiency. This is a problem of organization to make sure things do not fall between the cracks. For instance, there is no place for computer-aided design. It is done as part of projects where existing capabilities are used but hardly any new ones are developed. Hence, the way an organization is structured often establishes what the limits of the organization are.

AUTOMATION CROSS-CUT RESULTS

PRIORITY PROGRAM OFFICE NEEDS

OSTA - AUTOMATED DATA INTERPRETATION CAPABILITY

OSS - AUTOMATED MISSION OPERATIONS

OSTA - SATELLITE SERVICING CAPABILITY AUTOMATED LAUNCH CHECK-OUT

OSTDS - CONTINUED AUTOMATION OF TRACKING AND DATA ACQUISITION NETWORKS

CROSS-CUT RESULTS

CURRENT NASA AUTOMATION-RELATED DEVELOPMENT ACTIVITIES

TELEOPERATOR SENSORS AND CONTROL	OSS & OSTS AT JPL
SATELLITE SERVICING TECHNOLOGY	OSTS AT MSFC
MAN-MACHINE COMMUNICATION	OSTDS AT GSFC
PATTERN RECOGNITION AND IMAGE ANALYSIS	OSTA AT JSC
"NEEDS" DATA SYSTEM ARCHITECTURE AND SYSTEM DESIGN	OAST AT GSFC & JPL
LARGE SPACE STRUCTURE AUTOMATED ASSEMBLY AND CONTROL	OAST AT LaRC
AUTOMATED POWER SYSTEMS MANAGEMENT FOR SPACECRAFT	OAST AT JPL

CROSS-CUT RESULTS

RECENT AUTOMATION OF MAJOR OPERATIONS SUPPORT TASKS

STS FLIGHT DESIGN AND CREW ACTIVITIES PLANNING	JSC/OSTS
KSC LAUNCH PROCESSING SYSTEM	LSC/OSTS
TRACKING AND DATA ACQUISITION	GSFC/JPL/OSTDS
STS FLIGHT ASSIGNMENT MANIFEST	JSC/OSTS

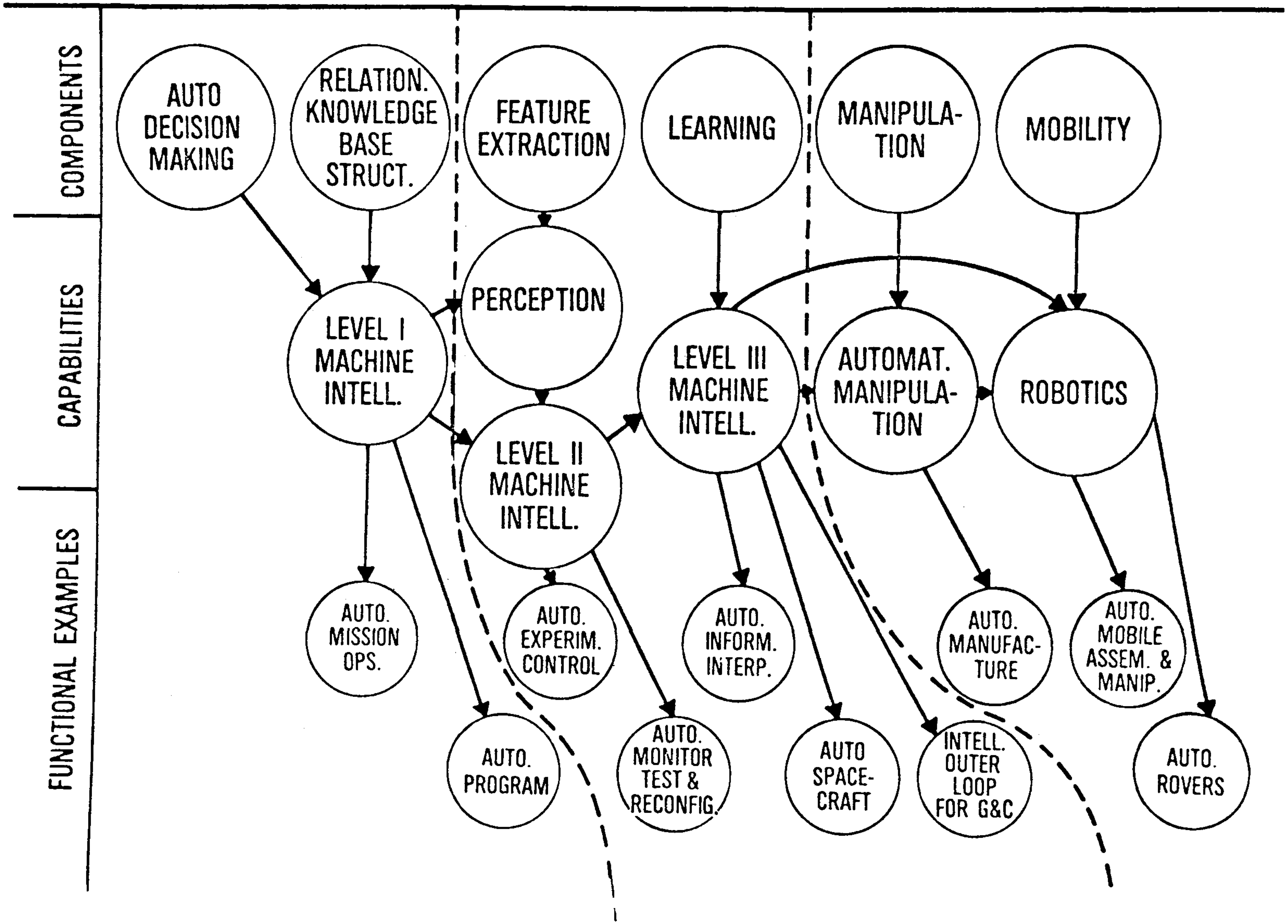
AUTOMATION OPPORTUNITIES

- REVOLUTION IN COMPUTER CAPABILITY
- INTENSIVE SPACE EXPLORATION AND UTILIZATION REQUIRES HIGH DEGREE OF AUTOMATION
- NSF/DOD/ARPA SUPPORT HAS ESTABLISHED RESEARCH BASE
- MISSION OPERATIONS SUPPORT IS COST EFFECTIVE AUTOMATION AREA:
  - -APPROXIMATELY 30% OF TOTAL PROGRAM COSTS
  - -CAPABILITIES CAN BE INTRODUCED AS AVAILABLE
  - -SPACE QUALIFICATION IS NOT AN ISSUE
- TELEFACTOR SYSTEM IS NEEDED FOR SPACE OPERATIONS SUPPORT

AUTOMATION PLAN OVERVIEW

- DETAILED ANALYSIS OF USER APPLICATIONS
- AUTOMATION OF OPERATIONS SUPPORT FUNCTIONS; EVOLVE TO ON-BOARD CAPABILITY
- DEVELOPMENT OF GENERIC TELEFACTOR TEST VEHICLE
- GENERIC TECHNOLOGY PROGRAM LEADING TO MACHINE INTELLIGENCE CAPABILITY

AUTOMATION
R&T BASE PROGRAM



PROPOSED APPROACH TO DEVELOPING AN AUTOMATION TECHNOLOGY BASE

1. DETERMINE THE STATE-OF-THE-ART ACROSS THE ASSOCIATED BREADTH OF TECHNOLOGY

   - WHAT TECHNOLOGY EXISTS
   - ITS CAPABILITIES
   - WHO HAS IT
   - HOW NASA MAY GAIN ACCESS TO IT
   - THE FUTURE CAPABILITY LEVEL OF TECHNOLOGY

2. ABSTRACT FROM EXISTING MODULES OF TECHNOLOGY HIGHER LEVEL ORGANIZATIONAL CONCEPTS THAT ARE EASIER TO GRASP AND UTILIZE

3. DEVELOP SELECTIVE IN-HOUSE CAPABILITY

4. PROVIDE A MODULAR TECHNOLOGY BASE FROM WHICH APPLICATIONS CAN BE ASSEMBLED

5. CONDUCT SELECTED SYSTEM DEMONSTRATIONS IN IMPORTANT POTENTIAL APPLICATION AREAS (RS/RT)

6. COORDINATE PROGRAM WITH OTHER AGENCIES

A FRAMEWORK FOR AUTOMATED DECISION MAKING AND PROBLEM SOLVING

EWALD HEER
UNIVERSITY OF SOUTHERN CALIFORNIA

In the context of this Presentation, a problem is a stated or perceived desire of one or more human beings to accomplish an objective.

For example, to keep a room's temperature at a certain level, or to drive to the airport, or to play and win a chess game, or to build a bridge, are such objectives.

Problem solving is the process of finding a way or means towards accomplishing a desired objective.

At the highest level, the problem solving process can be subdivided into two distinguishable phases which include the process of planning and execution.

Planning is an objective or goal oriented process of preparing a set of decisions, from alternative options, for action in the future.

The process of planning consists of objective identification, data acquisition and forecasting, decision-making -- where the process of decision-making involves synthesis, modeling and deductive activities. This requires the selection of a plan, possibly out of many feasible ones.

It requires the acquisition and deployment of operational resources, and it requires the control of the process when it is executed.

Control consists of three major elements: sensing or perception, deciding or reacting, and correcting, to stay within given limits.

PROBLEM SOLVING

- PLANNING - A GOAL ORIENTED PROCESS OF PREPARING A SET OF DECISIONS FROM ALTERNATIVE OPTIONS FOR ACTION IN THE FUTURE
  - OBJECTIVE IDENTIFICATION
  - DATA ACQUISITION AND FORECASTING
  - DECISION-MAKING
    - SYNTHESIS
    - MODELING
    - DEDUCTION
- EXECUTION - A GOAL ORIENTED PROCESS OF SELECTING AND IMPLEMENTING A SET OF DECISIONS WITHIN GIVEN CONSTRAINTS
  - SELECTION OF PLAN
  - OPERATIONAL RESOURCES ACQUISITION AND DEPLOYMENT
  - PROCESS CONTROLLING
    - PERCEPTION
    - REACTION
    - CORRECTION

We now look at some characteristics of planning and execution.

Two extreme cases with a continuous spectrum of possibilities in between can be identified.

First, the problem solver can identify and oversee every step and decision point of a problem in sufficient detail to develop the entire plan first, and then execute it accordingly.

In this case, the planning horizon reaches the objective, because the problem solver has enough information at his command, either from experience or from records, to predetermine every step and decision point towards accomplishing the objective. For instance, driving to the airport can be planned in great detail beforehand, if the route has been experienced and remembered, or if an accurate map is available. However, if neither is available, and if our eyesight is very short or not existent, then our planning horizon is very short -- involving perhaps only a few steps at a time.

In the extreme case, we have then an example of the second situation, where the problem solver can identify and oversee only one step and one decision point at a time towards a stated goal.

After each step, he must collect additional information or chance the risk of solving the wrong problem or no problem at all. In this second case, problem solving usually involves a great deal of search activity of one sort of another.

Humans have handled problem solving and decision-making tasks for millenia, although not always efficiently, to be sure. It is only during the last few decades that we have begun to acquire some scientific understanding of the processes by which humans solve problems and make decisions. We have learned that the complexity of the problem solving process, which makes its eventual outcomes (such as building a bridge, or going to the moon and planets) so impressive, is a complexity assembled out of relatively simple interactions among a large number of extremely simple basic elements.

We have shown how to synthesize thinking processes with computers that parallel closely the thinking processes of human subjects, in a substantial number of different problem solving tasks. The range of tasks that have been studied in this way is still narrow. However, there is little doubt that in this range, at least, we know what some of the principal processes of human thinking are, and how these processes are organized in problem solving programs.

We infer from these statements that the processes of problem solving and decision making can be automated, using digital computers, at least to a large degree. At a minimum, it will be possible to use the digital computer to amplify the human mental capabilities in problem solving and decision-making processes.

The computer is already proliferating in our society in management and industry to a degree unimagined only a decade ago. The computer is assuming more and more of the functions previously done by humans and is moving the boundary between manual and automatic control more and more towards increased automation. I believe that problem solving and decision-making tasks, if done automatically by computers, require programs developed by using techniques of machine intelligence, operations research, and advanced control theory.

PLANNING AND EXECUTION

- PLANNING HORIZON
- SEARCH ACTIVITIES
- PROBLEM SOLVING BY HUMANS
- PROBLEM SOLVING BY MACHINES
- PROBLEM SOLVING BY H-M-SYSTEMS
  - OPERATIONS RESEARCH
  - ADVANCED CONTROL THEORY
  - MACHINE INTELLIGENCE

In the following few vugraphs, I would like to put the subject of Automated Decision-Making and Problem Solving into the context of the NASA mission environment.

The NASA space activities can be subdivided into four major areas: Space Exploration, Global Services, Utilization of Space, and Space Transportation.

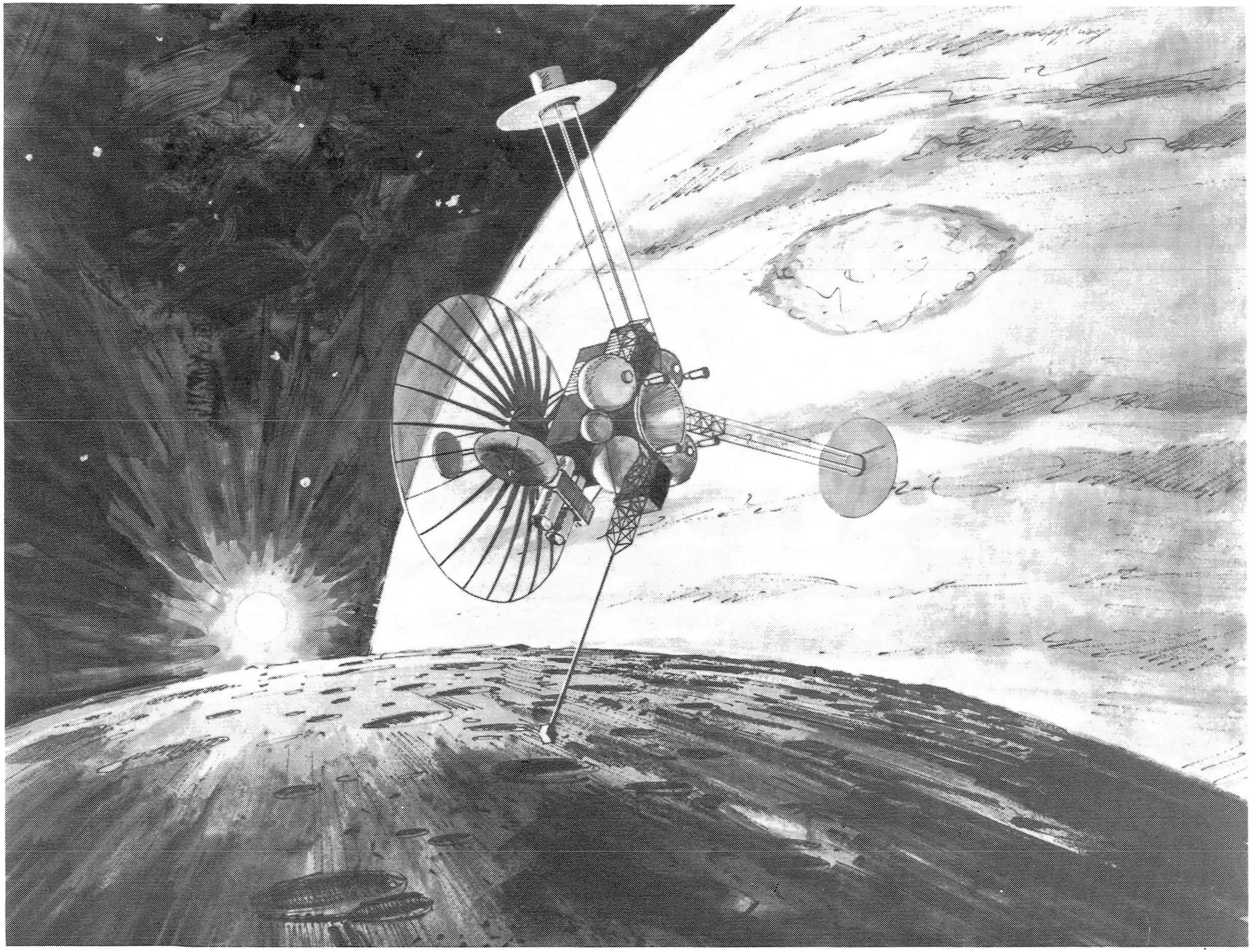
As the space program matures, its center of gravity is expected to move more and more towards utilization of space and space transportation. This brings with it similar or some of the same concerns of productivity, cost reduction, and cost effectiveness that are today already of such great concern to the ground-based industry.

It is within this environment that automated decision-making and problem solving technologies are expected to have their greatest potential impact.

Let us look at some of these systems by example in the designated order.

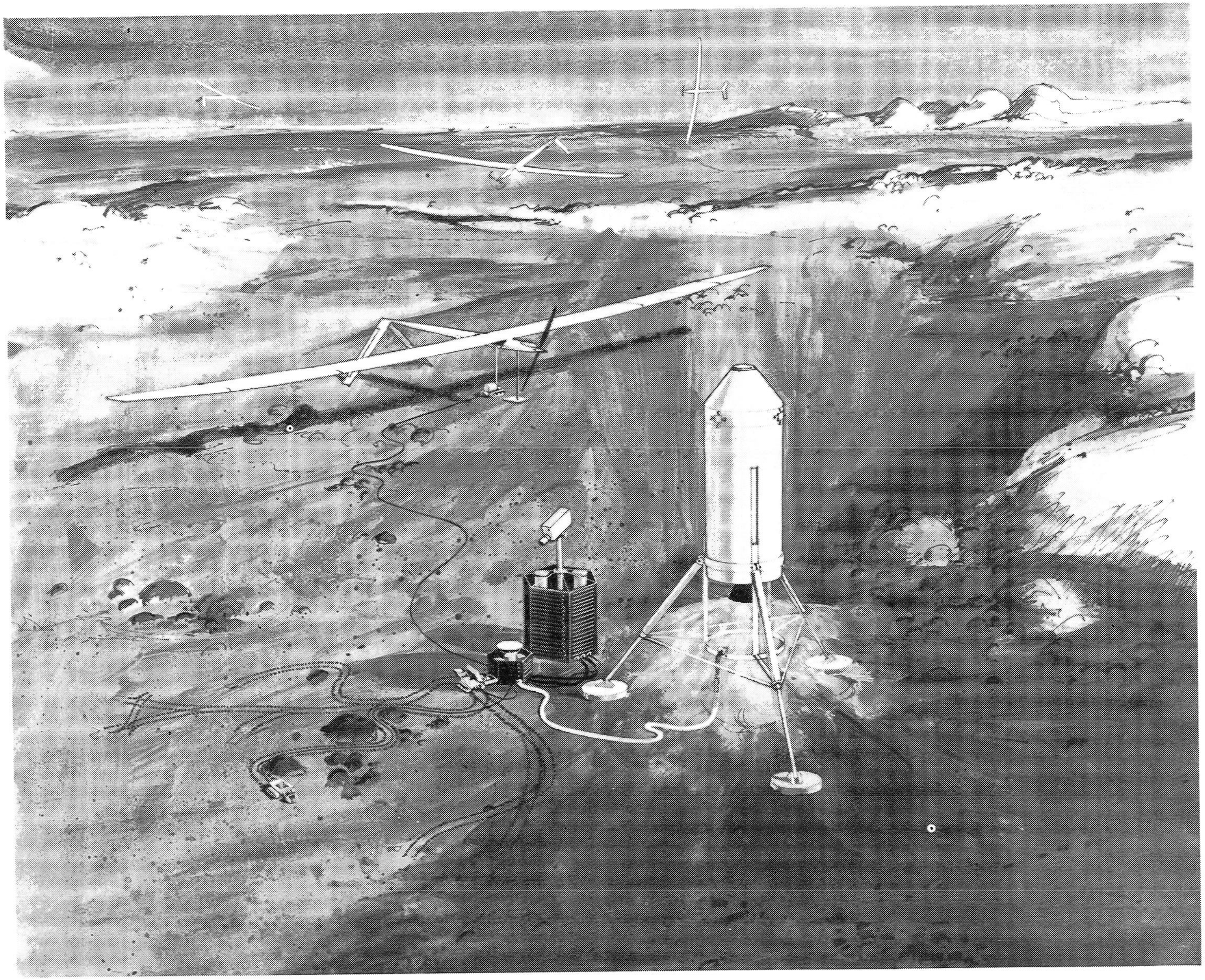
THE ENVIRONMENT OF NASA MISSIONS

- SPACE EXPLORATION
- GLOBAL SERVICES
- UTILIZATION OF SPACE
- SPACE TRANSPORTATION

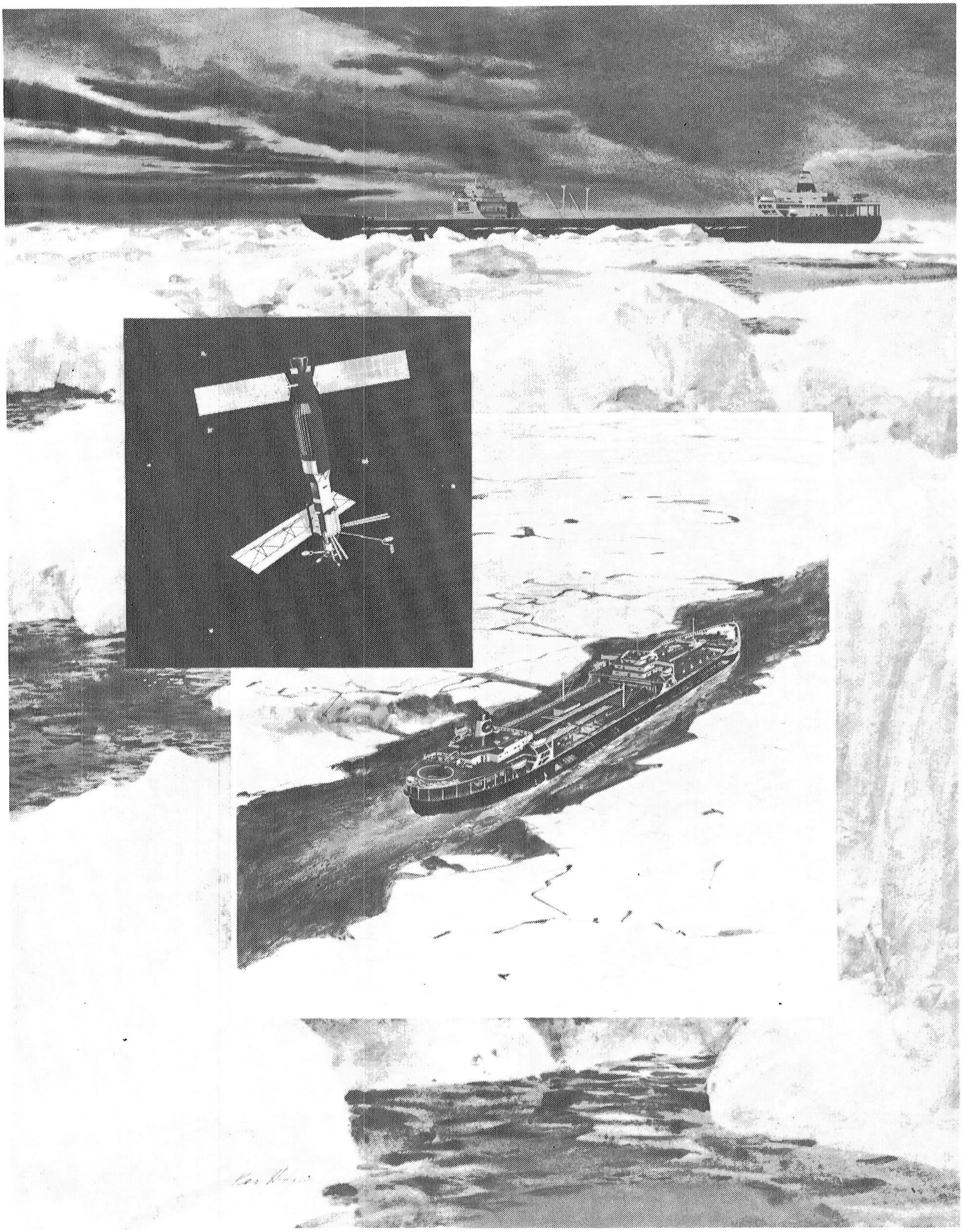


Galileo spacecraft navigates between Jupiter and Galilean satellites in rendering. After sending a probe into the Jovian atmosphere, the robot spacecraft will perform complex maneuvers at various inclinations with repeated close encounters with the satellites.

Mars surface robot will operate for two years and travel about 1000 km performing experiments automatically and sending the scientific information back to Earth.

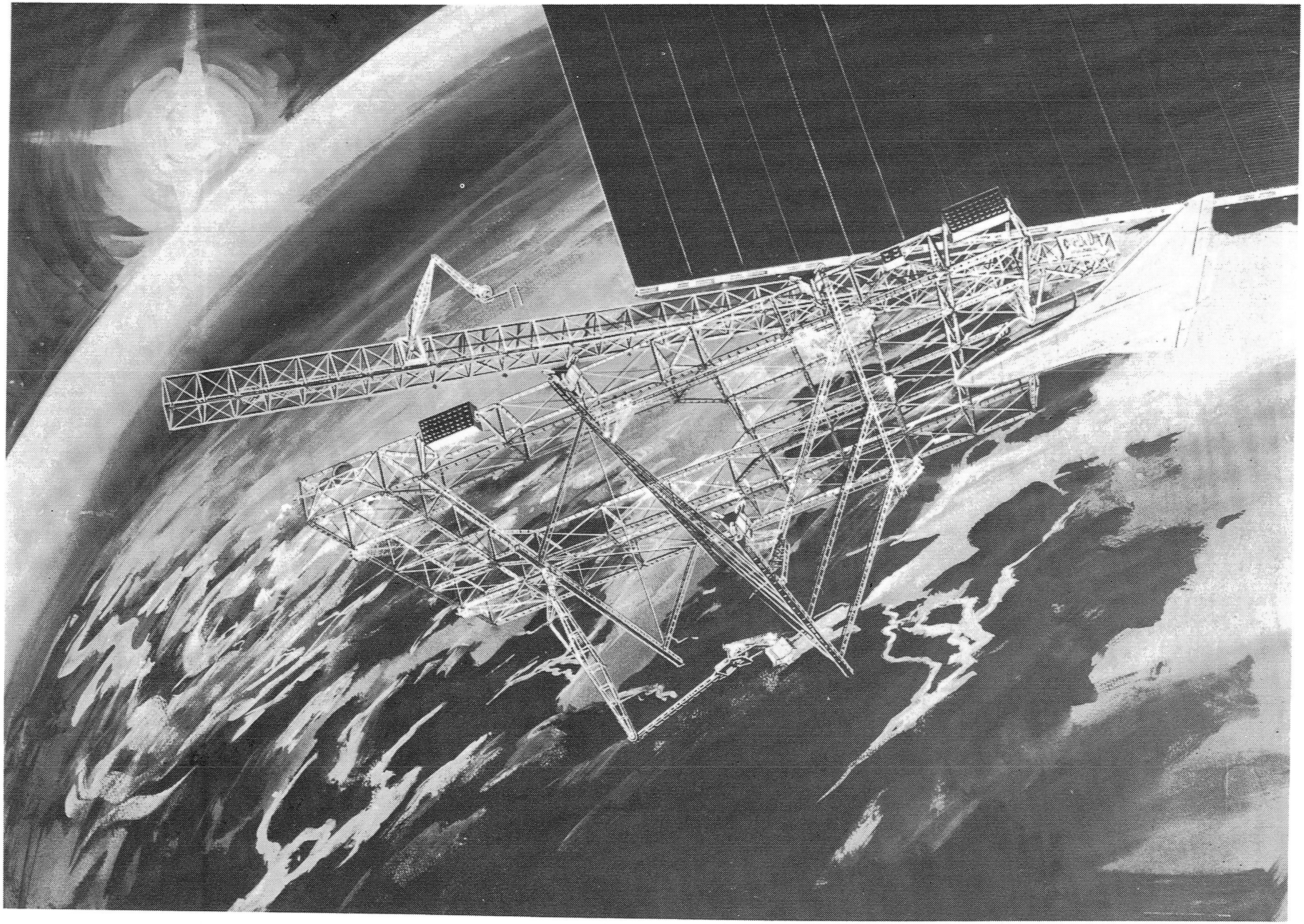


Artist's concept of a Mars surface scientific processing and sample return facility. Airplanes transport samples into the vicinity of the processing station. Tethered small rovers then bring the samples to the station for appropriate analysis and return to Earth.

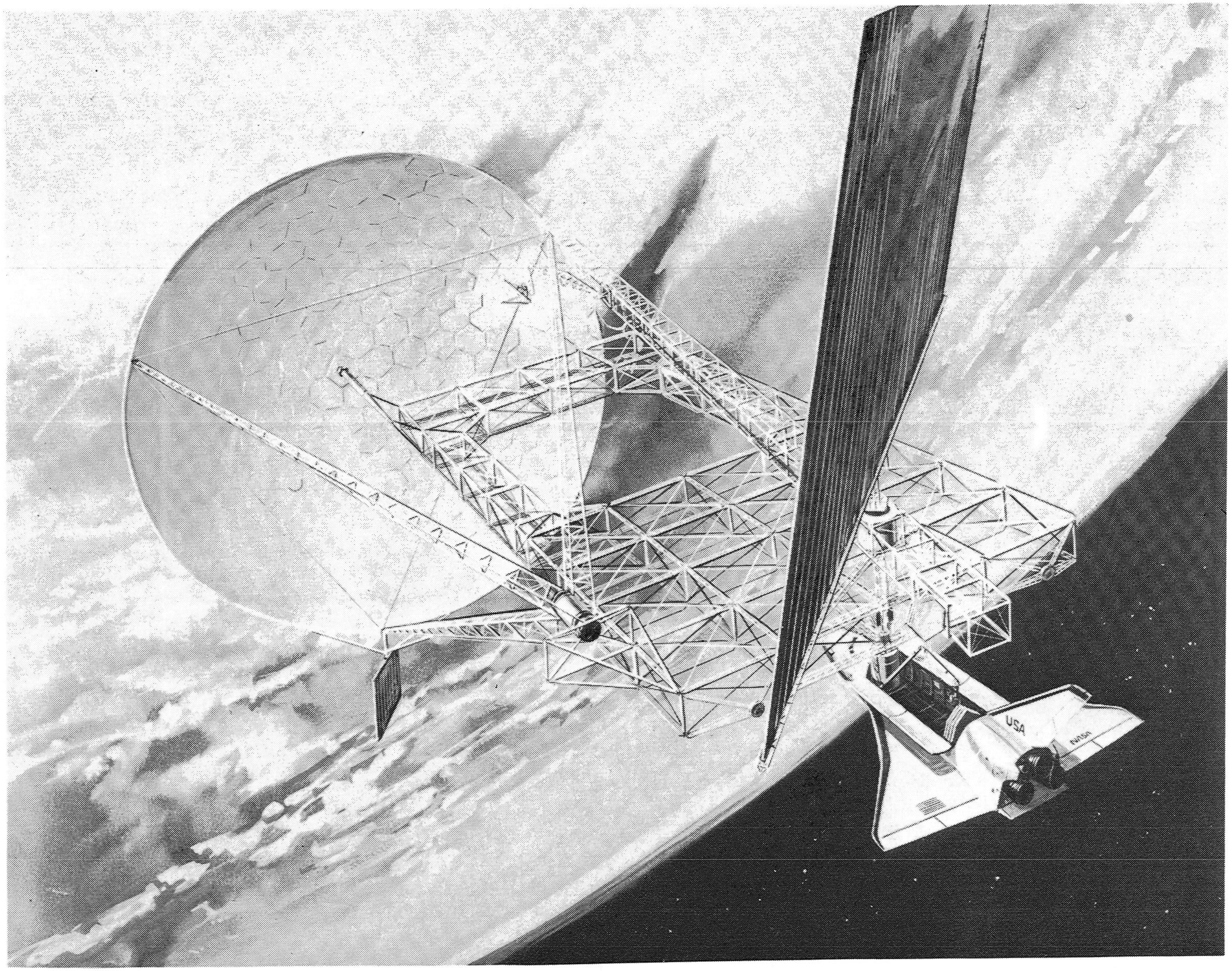


Seasat. The oceanographic satellite's high-data-rate Synthetic Aperture Radar imaging device has provided data on ocean waves, coastal regions, and sea ice.

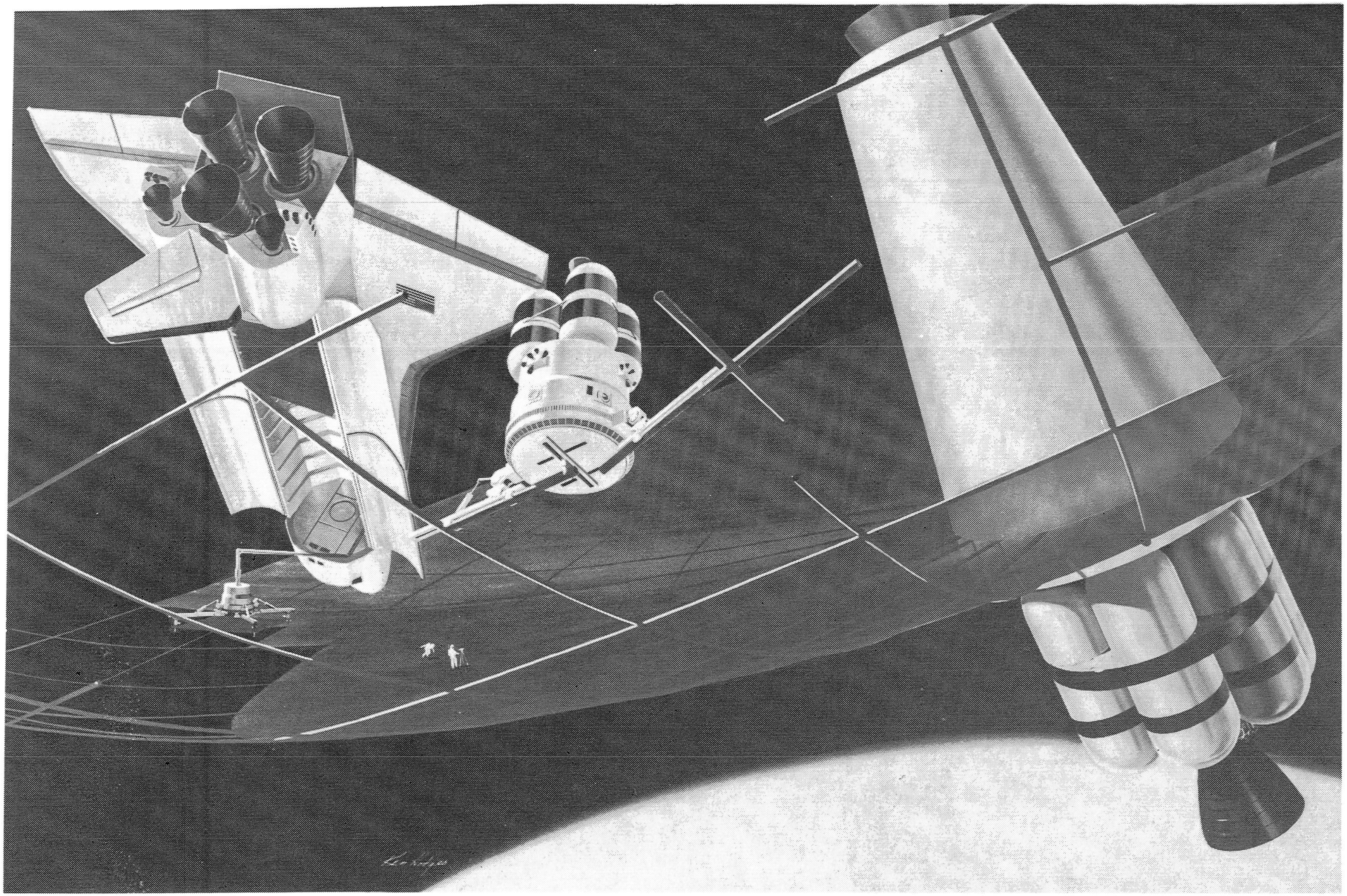
Large space systems require robot and automation technology for fabrication, assembly, and construction in space.



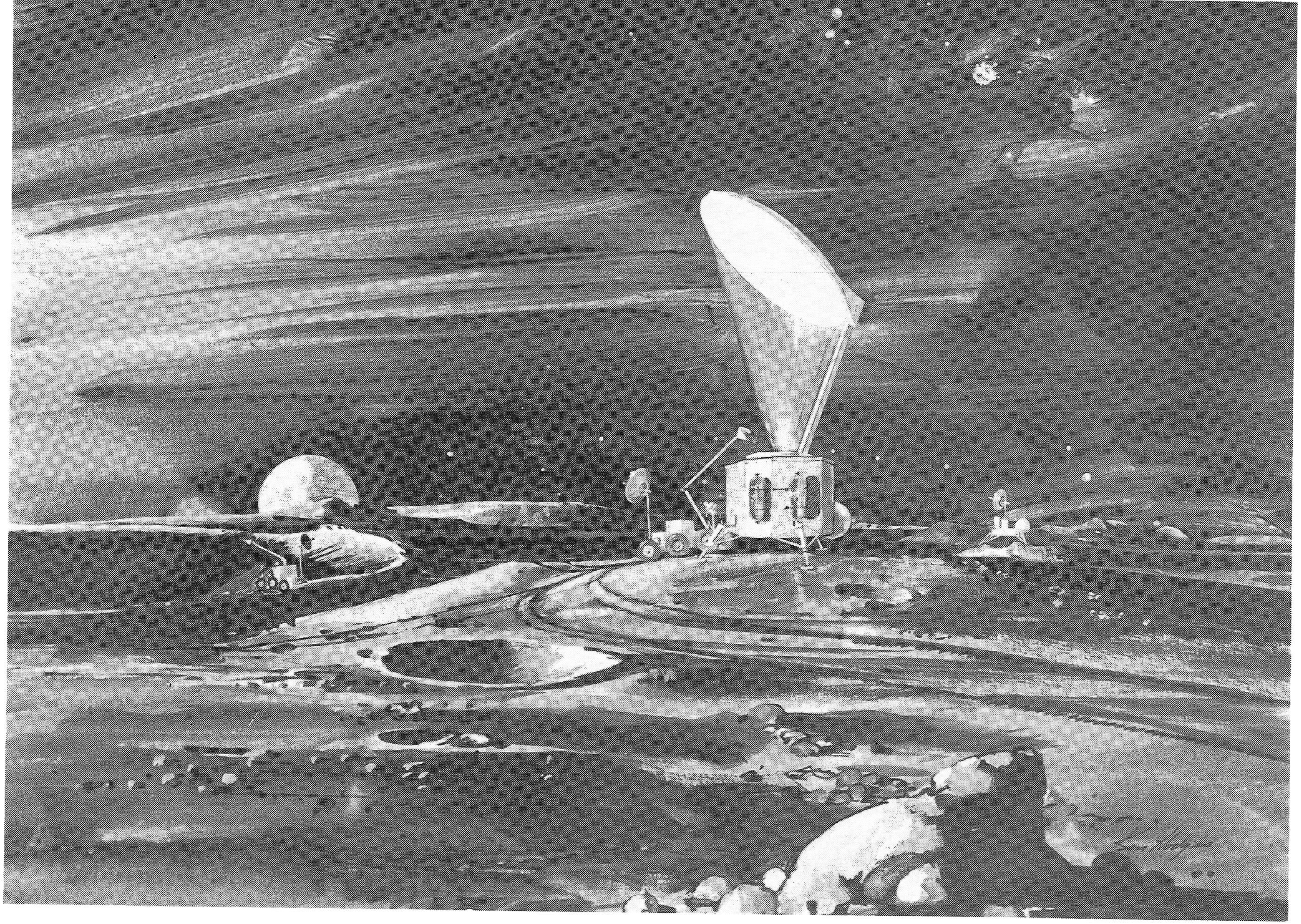
Complex construction facility in space with automatic beam builders, cranes, manipulators, etc., is served by the Shuttle.



Large space antennas are erected with the help of a space-based construction platform. The Shuttle brings the structural elements to the platform, where automatic manipulator modules under remote control perform the assembly.



Space construction of large antenna systems with automated tools, teleoperated manipulators, and free-flying robots.



Automated material processors on the lunar surface are serviced by robot vehicles with raw lunar soil.

The stated examples provide insight into the type of systems which are envisioned for the NASA space program, and which provide the content for the application of decision making and problem solving techniques.

Now let us see how we might establish a plausible connection between these areas.

There are two main phases a space system goes through: the Development Phase and the Operational Phase.

The development phase of a space system is not much different from the development phase of any product in industry on Earth as shown in this vugraph.

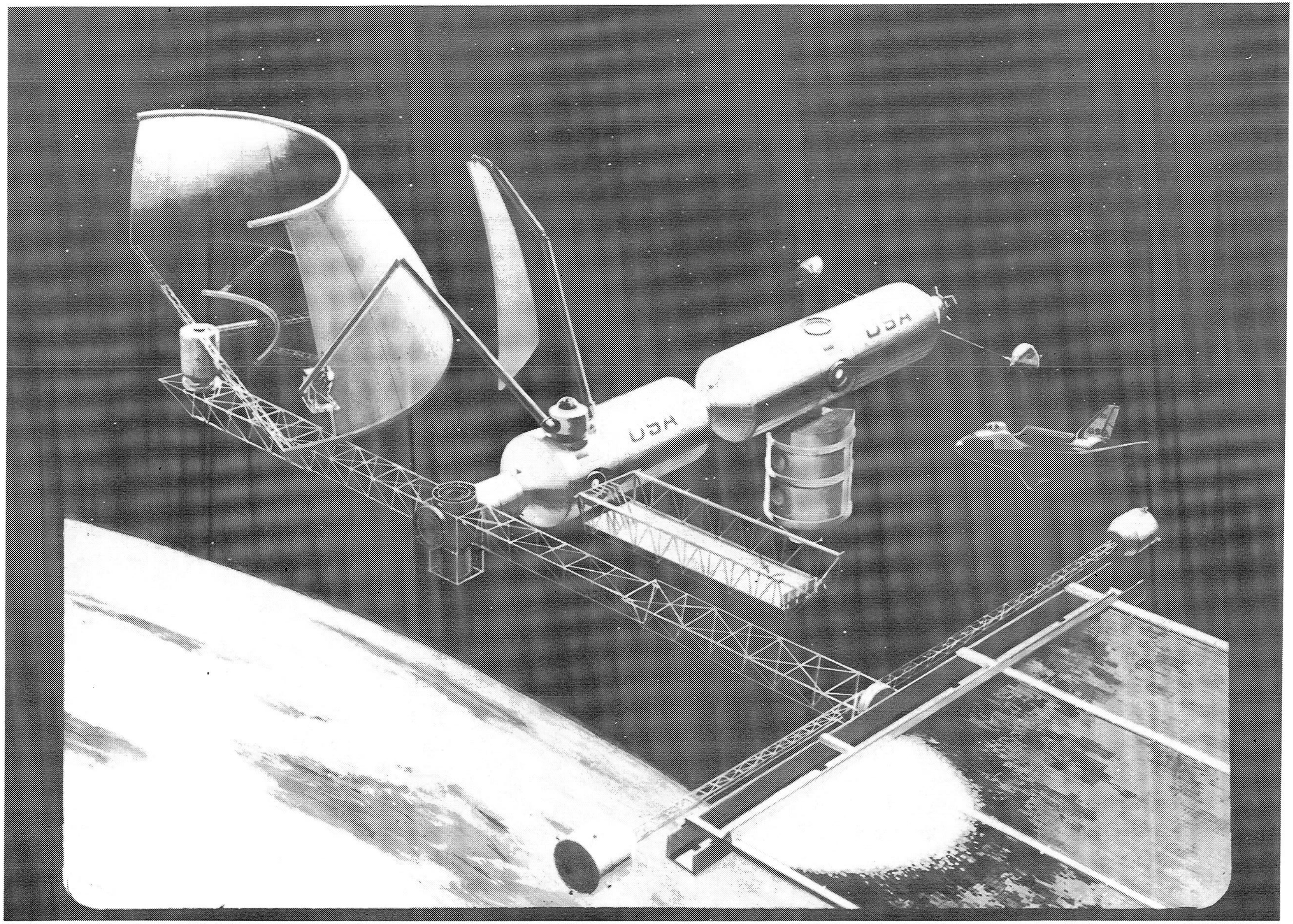
THE ENVIRONMENT OF SPACE

SYSTEM DEVELOPMENT

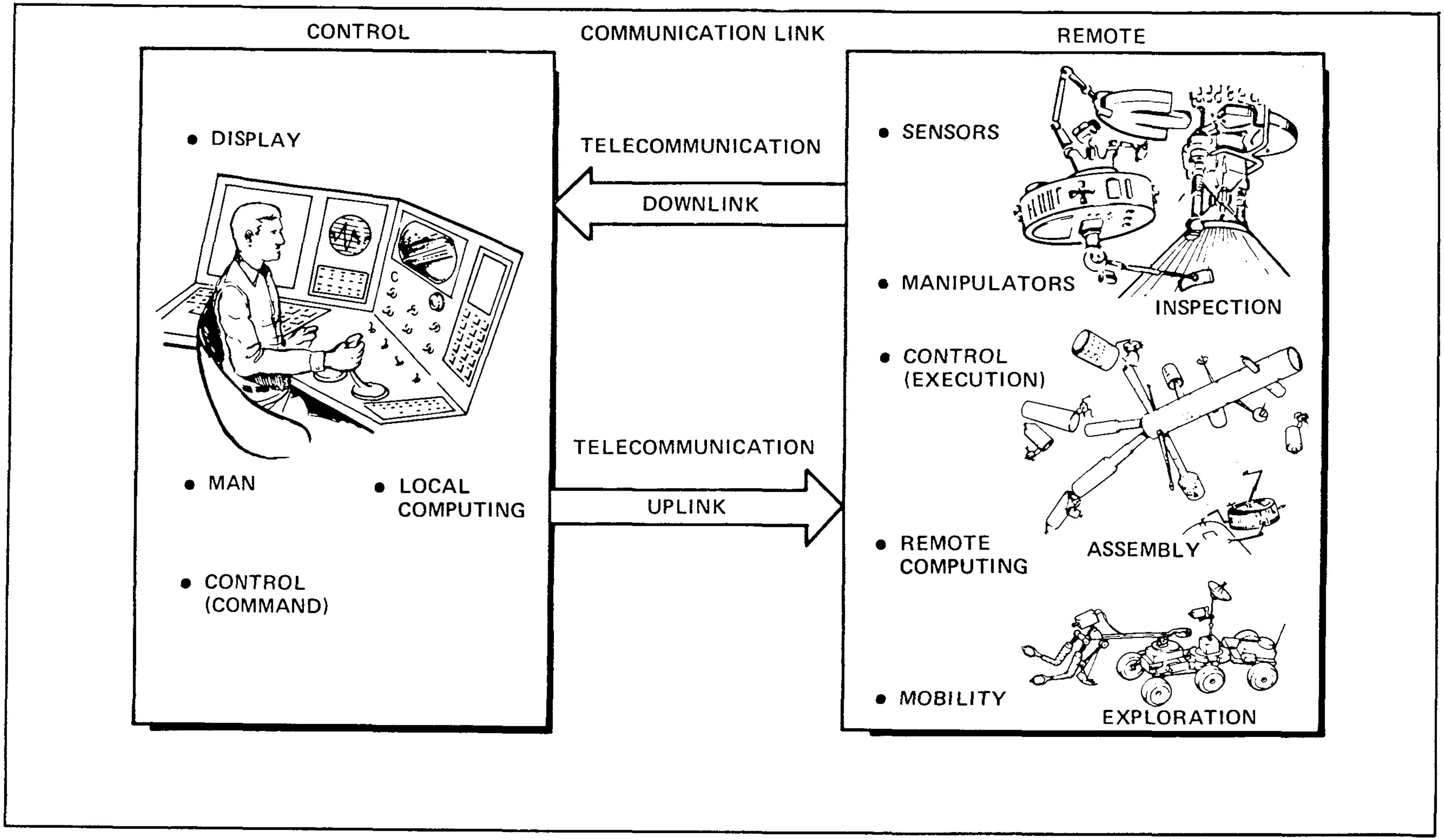
- PLANNING
  - CONCEPT
  - TECHNOLOGY
  - ENGINEERING
  - DESIGN
- EXECUTION
  - RESOURCES
  - CONSTRUCTION/PRODUCTION
  - TESTING
  - SHIPMENT

THE ENVIRONMENT OF SPACE SYSTEMS OPERATIONS

- ONBOARD SPACECRAFT OPERATIONS
  - NAVIGATION, GUIDANCE, AND CONTROL
  - RENDEZVOUS, STATION KEEPING, AND DOCKING
  - LANDING, TRAVERSING, AND ASCENT
  - DATA ACQUISITION AND PROCESSING
  - HOUSEKEEPING FUNCTIONS AND RESOURCE CONTROL
- IN-SPACE HANDLING AND ASSEMBLY
  - DEPLOYMENT, TRANSFER, ASSEMBLY AND JOINING
  - INSPECTION, SERVICING, MAINTENANCE AND REPAIR
  - RETRIEVING AND RESCUE
  - PROCESSING AND MANUFACTURING
  - MINING
- GROUND-BASED OPERATIONS
  - DATA INTERPRETATION, DISTRIBUTION, AND ARCHIVING
  - GOAL SELECTION, OPERATIONS PLANNING, AND SEQUENCING
  - SIMULATION
  - MONITORING

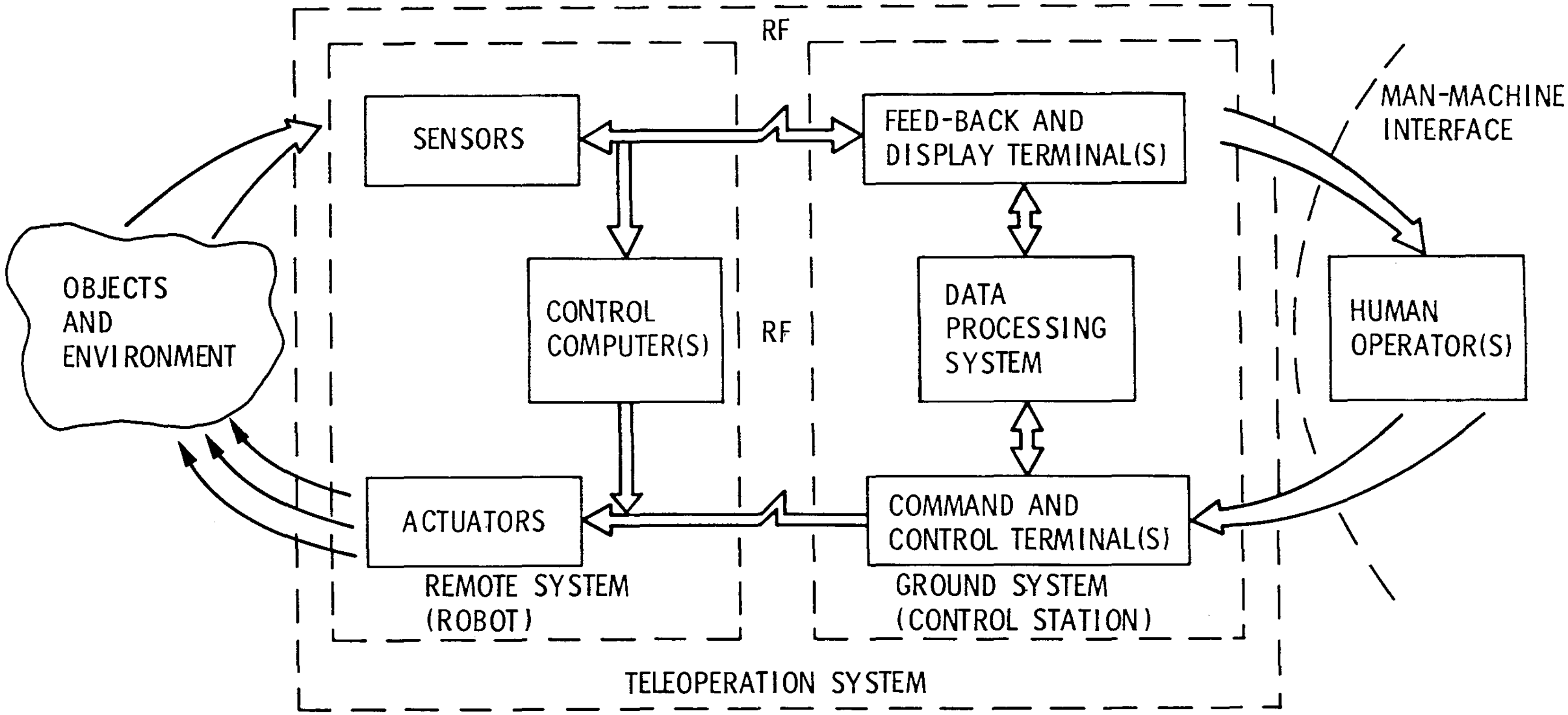


An example of a complex space operations facility is depicted here. It is the construction of a space station. Bulk material is brought by the Shuttle. Structural elements are fabricated at the construction facility and then assembled by remotely controlled manipulators.



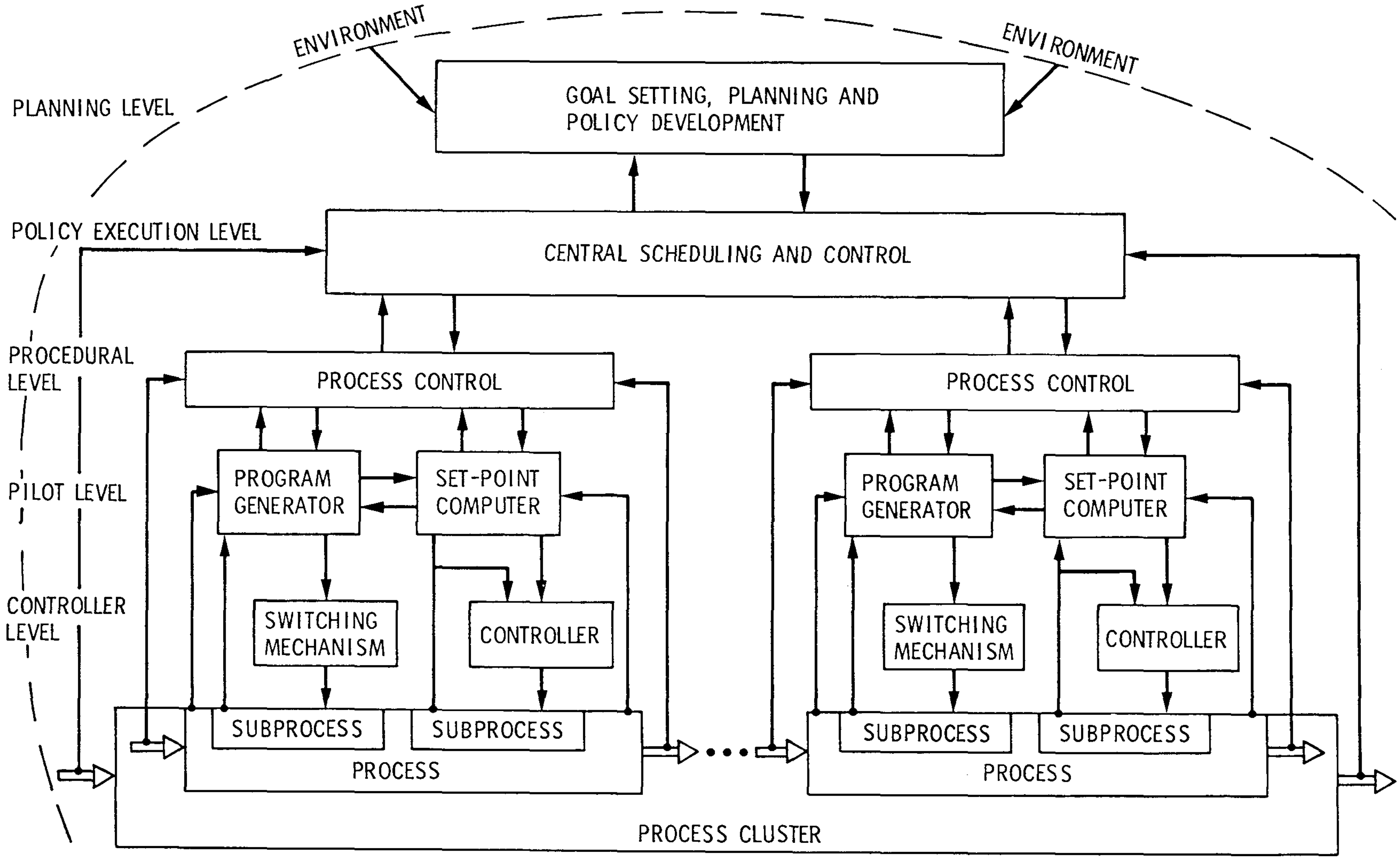
The concept of remote operation and control in space is shown here. The decision-maker in the control center is represented by one or more human operators. The remote system is controlled via telecommunication links directly, or, the remote system can make its own decisions depending on the autonomous capabilities built into it.

MISSION OPERATIONS MODEL



In a more abstract representation, the "teleoperation system" is between the human operator(s) and the objects and environments where the system's objectives are executed. The autonomy of the remote system increases, in general, as the control computer(s) capabilities grow in terms of decision-making and problem solving capabilities.

HIERARCHY OF OPERATIONAL SYSTEMS



A different cut of an operational system is given in this vugraph. It shows the inherent hierarchical structure of operational systems.

The "low level" solution concepts identified here are generally well within the scope of existing technology. They do not pose a technological bottleneck.

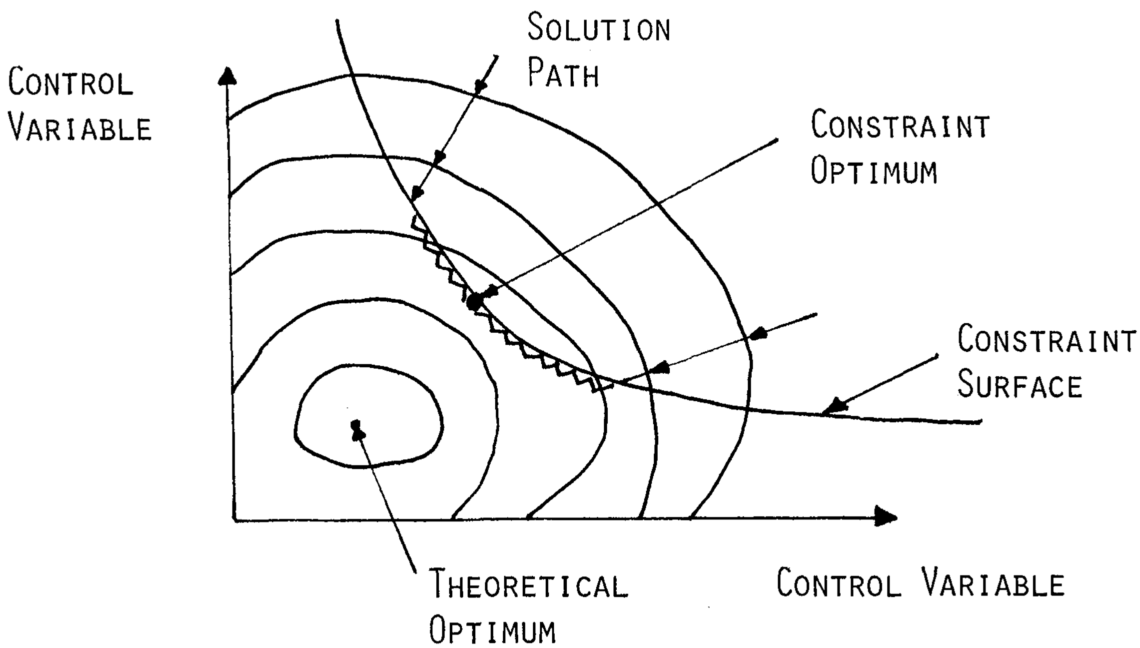
LOW LEVEL SOLUTION CONCEPTS

- ANALOG/DIGITAL (PID) CONTROLLERS
- DIRECT DIGITAL CONTROL
- CONTROL MODE SWITCHING
- FIXED CONTROL PROGRAMS

The "intermediate level" solution concepts are still under development. Although they are in principle understood, the practical effective application still needs R&D work in the area of system identification and complex system modeling.

INTERMEDIATE LEVEL SOLUTION CONCEPTS

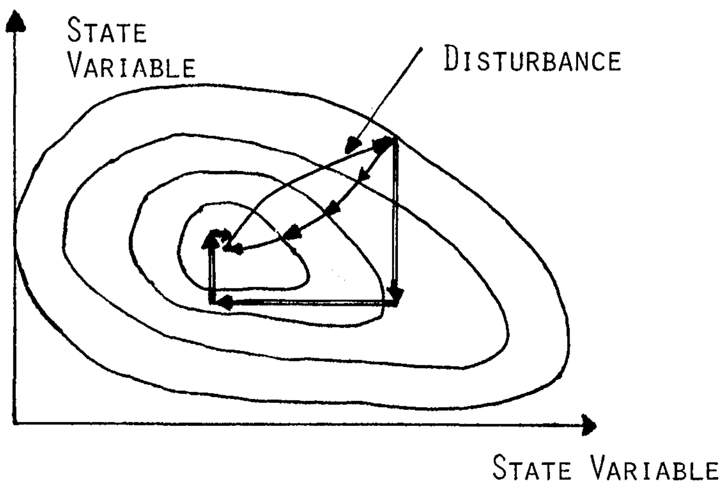
- CONTROL PROGRAMS WITH PROCESS DRIVEN PROGRAM SELECTION
- CONTROL PROGRAMS WITH PROCESS MODELING



The "high level" solution concepts require still the most extensive research and developments. They usually pertain to problems with ill-defined structures, not solvable by analytical mathematical approaches. They usually require some search approach. Some of the solution concepts incorporate the ability for the system to learn from past experiences (operations, sensor information, etc.). This is illustrated in the figure. If the system is disturbed from optimal performance, it may, e.g., return to optimal performance as shown by the double lined path. If the system is disturbed in a similar fashion, it may have learned a better way to return to optimal performance, namely by the single lined path.

HIGH LEVEL SOLUTION CONCEPTS

- CONTROL PROGRAM WITH SEARCH OPTIMIZATION AND LEARNING CAPABILITY
  - STATISTICAL SEARCH
  - LINE SEARCH
  - UNIVARIATE SEARCH
  - COMPLEX METHOD
  - GRADIENT TECHNIQUES
  - HEURISTIC TECHNIQUES



Problems can be subdivided into two main categories: well-structured problems, and ill-structured problems. The first require routine repetitive decisions which are generally amenable to programmable decision processes. The second require novel nonprogrammable decision processes.

The decision-making processes can be subdivided into those representative of those done by humans and those done by machine. Many of such decision processes require a combination of humans and machines.

PROBLEM STRUCTURE AND SOLUTION TECHNIQUES

- WELL-STRUCTURED PROBLEMS
  - ROUTINE REPETITIVE DECISIONS
  - PROGRAMMABLE DECISION PROCESSES
- ILL-STRUCTURED PROBLEMS
  - NOVEL POLICY DECISIONS
  - NONPROGRAMMABLE DECISION PROCESSES

- HUMAN ORIENTED DECISION-MAKING
  - HABIT
  - CLERICAL ROUTINE
  - STANDARD PROCEDURES
  - WELL-DEFINED COMMUNICATION CHANNELS
  - JUDGEMENT
  - INTUITION AND CREATIVITY
  - RULES OF THUMB
  - SELECTION AND TRAINING OF MANAGERS

- AUTOMATION ORIENTED DECISION-MAKING
  - OPERATION RESEARCH
  - COMPUTER DATA ANALYSIS AND PROCESSING
  - HEURISTIC PROBLEM SOLVING
  - HEURISTIC COMPUTER PROGRAMS

INTELLIGENT CONTROL SYSTEMS

George N. Saridis
Purdue University

INTELLIGENT CONTROLS FOR ADVANCED AUTOMATED PROCESSES

G. N. Saridis
Purdue University

ABSTRACT

This paper deals with the evolution of ideas of intelligent controls and their application to high level man-machine interactive systems like general purpose manipulators, industrial robots, prosthetic devices for amputees, and orthotic devices for paralyzed persons. Some case studies are presented to demonstrate the feasibility of the approach.

1. Introduction

The phenomenal technological achievements of the past fifty years must be credited to a large extent to the development of the system theoretic methodologies. These methodologies are based on a rigorous mathematical modeling and subsequent analysis of the associated physical process, a systematic synthesis of precise controls resulting in the design and effective operation of industrial, economic, urban and even space exploration systems that have become essential parts of the socioeconomic environment of the modern societies [27,38]. As the world economy is reaching a turning point due to the depletion of certain popular energy resources while higher demands are imposed from space exploration, work in hazardous environments, modernization of industrial plans, and efficient transportation of large groups of people, new methodologies are developed suitable for computer utilization and demonstrating advanced machine intelligence and decision making [40]. For over twenty years scientists have been developing the cognitive field of Artificial Intelligence, more or less in the image of a human brain. Significant results have been accomplished in Speech Recognition, Image Analysis and Perception, Data Base Analysis and Decision-Making, Learning, Theorem Proving and Gains, Autonomous Robots, etc. [1,8,9,13,14,15,16,20,21,25,28,29,30,32,34,35,37,49,50,51,54,55]. The discipline that couples these advanced methodologies with the system theoretic approaches necessary for the solution of the current technological problems of our societies is called "Intelligent Controls" [12,40].

Intelligent Controls utilizes the powerful high-level decision making of the digital computer with advanced mathematical modeling and synthesis techniques of system theory to produce a unified approach suitable for the engineering needs of the future.

One of the most important applications of Intelligent Control Theory is in manipulative systems. These systems may involve the control of a general purpose manipulator for space exploration, like the Mars-rover, or a hazardous environment robot for operation in a nuclear containment, or a hospital aid manipulator, an electrically driven prosthetic limb to replace an amputated arm or even an orthotic brace to assist paralyzed people [6,10,11,22,24,25,26,41,44,47,52]. Such devices impose special considerations and constraints in terms of small weight, small physical dimensions, real time performance, human limb appearance and functionality, and most restrictive, a small number of non-interacting command sources, i.e., of a command vocabulary, and a small number of sensors. The above constraints exclude computationally complex algorithms or long computation time. Also, training of the operation to generate combinatorial command codes must be very limited. Hence, such systems must maximize flexibility of performance subject to a minimal input dictionary and minimal computational complexity.

The paper will consider the general theory of Intelligent Controls first and then its application to general purpose robots, prostheses and orthotic devices.

2. Cognitive Systems and Intelligent Controls

Cognitive systems have been traditionally developed as part of the field of artificial intelligence to implement, on a computer, functions similar to the ones encountered in human behavior [1,12,28,33,37,54,55]. Such functions as speech recognition and analysis, image and scene analysis, data base organization and dissemination, learning and high level decision making, have been based on methodologies emanating from simple logic operations to pattern recognition and linguistic and fuzzy set theory approaches [56]. The results have been well documented in the literature.

In order to solve the modern technological problems that require control systems with intelligent functions such as simultaneous utilization of a memory, learning, or multilevel decision-making in response to "fuzzy" or qualitative commands, a new generation of control systems has been developed. They are named "Intelligent Controls" and utilize the results of cognitive systems' research effectively with various mathematical programming control techniques [27,38]. Each cognitive system associated with the specific process under consideration may be considered as subtask of the process requested by an original general qualitative command, programmed by a special high-level symbolic computer language, and sequentially executable along with decision making and control of the hardware part of the process.

Many systems have been designed to perform in the above manner. In the area of manipulators and robotics many such systems have been developed for object handling in an industrial assembly line, remote manipulation in hazardous environments, the planet-exploration Mars-vehicle, hospital aids to disabled, and autonomous robots [2,3,4,11,24,25,26,33,36,42,53]. In most cases where the control process is remotely performed from the operator, its function is semi-autonomous and the system must utilize some cognitive systems to understand the task requested to execute, identify the environment and then decide for the best plan to execute the task.

Various pattern recognition, linguistic or even heuristic methods have been used to analyze and classify speech, images or other information coming in through sensory devices as part of the cognitive system [1,5,6,23,28,37]. Decision-making and motion control were performed by a dedicated digital computer using either kinematic methods, like trajectory tracking, or dynamic methods based on compliance, dynamic programming or even approximately optimal control [45].

A Hierarchically Intelligent Controls approach has been proposed by Saridis as a unified theoretic approach of cognitive and control systems methodologies. The control intelligence is hierarchically distributed according to the principle of "Decreasing Precision with Increasing Intelligence," evident in all hierarchical management systems [40]. It is composed of three basic levels of controls even though each level may contain more than one layer of tree-structured functions:

- The Organization level
- The Coordination level
- The Hardware Control level

The organization level is the master mind of such a system. It accepts and interprets the input commands and related feedback from the system, defines the task to be executed and segments it into subtasks in their appropriate order of execution. An appropriate subtask library and a learning scheme for continuous improvement provide additional intelligence to the organizer. Since the organization level takes place on a medium to large size computer appropriate "translation and decision-making schemata" are linguistically implementing the desirable functions [41].

The coordination level receives instructions from the organizer and feedback information from the process for each subtask to be executed and coordinates the execution at the lowest level.

A lowest-level control process usually involves the execution of a certain motion and requires besides the knowledge of the mathematical model of the process the assignment of end conditions and performance criteria or cost functions. The coordinator, usually composed of a decision-making automaton representing a context free language, may assign both the performance index and end conditions as well as possible penalty functions designed to avoid inaccessible areas in the space of the motion. The decisions of the coordinator are obtained with the aid of a performance library and a learning decision-making scheme updated to minimize the cost of operation. Optimal or approximately optimal control system theory may be used for the design of the lower level controls of decentralized subprocesses of the overall process to be controlled.

The method has been successfully applied to the control of a general purpose manipulator with visual feedback and voice inputs [42], and a traffic control system for an integrated urban and highway environment [43]. The next section describes a hierarchically intelligent control system for a prosthetic arm.

3. A Hierarchically Intelligent Control for a Prosthetic Arm

An electronically controlled, electrically driven prosthetic arm is an excellent case for application of the hierarchically intelligent control method. It has been found that a small number of non-interacting commands can be easily retrieved conveniently from myoelectric signals [23,46]. These can serve to activate an artificial limb or a powered orthotic device in up to 6 degrees of freedom. Presently, we have successfully activated 6 functions using double electrode location, within adequate time. Anything slower is again unacceptable to the user. Using pattern classification or nonlinear signal identification, one can discriminate between up to some 30 different commands from two electrode sites [46]. More sites are prohibitive since they hardly exist in high level amputees or in paralyzed persons who are the ones who will most need a robotic arm. Also the more sites, the more you constrain the freedom of movement of the user and the higher the probability of failure. Furthermore, single sensors, lightweight and which can be blended easily into the limb in its appearance without drawing attention and causing embarassment to the user, can also be incorporated. The simplest ones would be torque sensors and sensors to feed back to the system joint positions. Feedback to the nervous system would of course be best. However, apart from visual feedback, the progress here is unfortunately slow and cannot yet be considered as being concrete help.

A design for a p-degree of freedom prosthetic arm with a hierarchically intelligent controller, supervised and commanded by the brain has been presented in [47] to establish the feasibility of such a control system using EMG commands and some sensory feedback. The purpose is to construct a prosthetic upper limb for amputated persons or hazardous environment use, with anthropomorphic characteristics of motion, which would receive commands directly from the brain of the operator either through a direct tapping of the nervous system or through electromyographic signals of any other source of qualitative commands, resulting from the operator's survey of the environment. It is conjectured that such an anthropomorphic function of the arm, in response to crude qualitative commands, can be obtained only by a hierarchical, say, three-level control, with hierarchically increasing order of intelligence and hierarchically decreasing order of precision of control signals as one moves from the electromechanical actuators of the arm to the interface with brain. A block diagram in Fig. 1 illustrates the concept of such a hierarchically intelligent control system. The three levels of control are (1) a linguistic organizer, (2) a fuzzy automaton as a coordinator, and (3) a bank of self-organizing controls. They are described briefly below along with their function in the control system.

3.1 The Self-Organizing Control Level

The arm as a whole process may in general be subdivided into p subprocesses, one per degree of freedom, described by the following set of generalized differential equations:

$$\dot{x}_i = F_i x_i + B_i(\overline{x}_i)u_i + f_i(x,\dot{\overline{x}},w_i)$$

$$z_i = H_i x_i + v_i \qquad\qquad i = 1, s, \ldots, p$$

where x_i is the n_i-dimensional state vector, z_i is the r_i-dimensional output vector, and u_i is the m_i-dimensional control vector of the ith subprocess; F_i, $B_i(\overline{x}_i)$, and H_i are matrices of appropriate dimension, w_i and v_i appropriate noise vectors, and $f_i(\cdot)$ nonlinear functions representing the gravity influence and the coupling terms from other subsystems through various reaction forces. If the subsystem was isolated from those force fields, these terms would be zero and the system would be linear in x_i. Furthermore, the state and the output of the overall system are defined respectively by:

$$x^T = [x_1^T, \ldots, x_p^T], \quad z^T = [z_1^T, \ldots, z_p^T]$$

$$\bar{x}_i^T = [x_1^T, x_2^T, \ldots, x_{i-1}^T, x_{i+1}^T, \ldots, x_p^T],$$

A performance criterion for the proper mechanical function of the system may be defined as

$$J = \sum_{i=1}^{p} \mu_i J_i(\alpha_i)$$

where $J_i(\alpha_i) i = 1,..p$ are appropriate performance criteria, α_i are adjustable coefficients relative to the speed of response of the subprocesses, and μ_i are adjustable coefficients $\mu_i = 0,1, i = 1,\ldots,p$, relative to the appropriate blending of primitive motions of the subsystems to generate an appropriate compound motion for the arm. Obviously $J_i(\alpha_i)$ is the performance criterion for each subprocess, assumed to be of infinite duration for simplicity of implementation, a conjecture verified experimentally. A feedback control may be structured per subsystem as

$$u_i(z) = K_i \phi(z_i) + C_i \psi(z)$$

where the first part is the optimal control for the uncoupled subsystem, while the second term represents a nonlinear term depending on the nonlinear coupling with the other subsystems. The expanding subinterval algorithm may be applied to yield the asymptotically optimal coefficients K_i and C_i for each subprocess, thus creating performance-adaptive Self Organizing Controls for the lowest level of the hierarchically intelligent control system. Self-organization is necessary to avoid parameter identification of the system dynamics.

The higher levels are interfaced with the individual self-organizing controls through the adjustable relative speed coefficients α_i, the blending coefficients μ_i, and the desired final states $x_i^d(T)$.

From the preceding discussion, it is obvious that the third-level Self Organizing Controls are designed for precision control of the mechanical subprocesses at hand but do not exhibit higher-quality intelligent functions, such as intelligent decision-making for motion coordination, direction, and goal accomplishment. With their limited capabilities, they resemble more the reflexes of a biological system. The higher intelligent functions are hierarchically distributed to the higher levels of the controls which are described next.

3.2 <u>The Fuzzy Automaton as a Control Coordinator</u>

A <u>fuzzy automaton</u> has been developed as an extension of the variable structure stochastic automata to accommodate inputs of a "fuzzy" nature defined by Zadeh [56].

A fuzzy quantity belongs to a set of values describing an "object" with undefined boundaries that are characterized individually by a membership coefficient to the set.

A fuzzy automaton is therefore defined as a sextuple $[Z,Q,U,F,H,\zeta]$, where, in addition to the finite set of "fuzzy" inputs $Z = \{z\}$, finite set of states $Q = \{q\}$, finite set of outputs $U = \{u\}$, state transition function F, and output function H, a fuzzy membership vector ζ is assigned to the states of the automaton [19]. A membership transition matrix $\Xi^k(n)$ for each fuzzy input z^k may be assigned to update the membership functions ζ.

Such a fuzzy automaton may be used to coordinate the p-primitive motions of the subprocess to form a desired compound movement of the arm from an initial state $x_0(t_0)$ to a final predefined state $x^d(T)$. Such a coordination is needed to put together the right amount of primitive motions and their proper velocities in order to accomplish the proper compound motion in response to a fuzzy command of the higher-level intelligent controller which is not to be bothered with the details of coordination. A fuzzy automaton $[C,Q,Q,F,I,\zeta]$ has a structure natural for such a coordinator, if $C = \{c\}$ is the set of fuzzy command inputs transmitted from the organizer, and $Q = \{q\}$ is the set of the states as well as outputs of the automaton representing the appropriate performance criteria of each of the subprocesses assigned to generate the appropriate motion.

3.3 Linguistic Methods for Intelligent Organization

Based on the preceding discussion, given a command and a terminal state, the fuzzy automaton can be trained to produce the proper compound motion for the arm, which represents a level of control more intelligent than the self-organizing. However, the commands generated from the brain of the human operator are more in the form of compound tasks, like picking up a glass of water to drink. Therefore an intelligent control system is needed to interface the brain with the fuzzy automaton and translate the above qualitative command to a sequence of compound motions of the area that will accomplish the task. Such a control will be required to produce a segmentation of the task, providing appropriate $x^d(T)$ and qualitative information of the compound motion of the arm for each segment. It should produce on-line information about the change of direction, combination, or expansion of segments, evaluation of the accomplishment of the task and processing of sensory feedback information from the brain, etc., without burdening the operator with unnecessary details about the function of the arm.

A machine producing decisions and functions of such a high level of intelligence must be an advanced digital computer, capable of processing qualitative information of high content, but also of fuzzy nature in the sense that high precision in execution is not required. A natural system for this type of information processing is the previously mentioned linguistic methods approach which has been developed in the modern literature for artificial intelligence, pattern-recognition, scene analysis, and other functions [15]. Such methods process strings of words with logic instructions to accomplish the task according to certain predetermined grammar and syntax in manner similar to natural languages. In particular, stochastic grammars developed by Fu [15] for syntactic pattern recognition or fuzzy grammars proposed by Zadeh [56] are most desirable for a generation of the command strings appropriate to organize the motions of the artificial arm. The reader is referred to Ref. [47] for detailed information on this project.

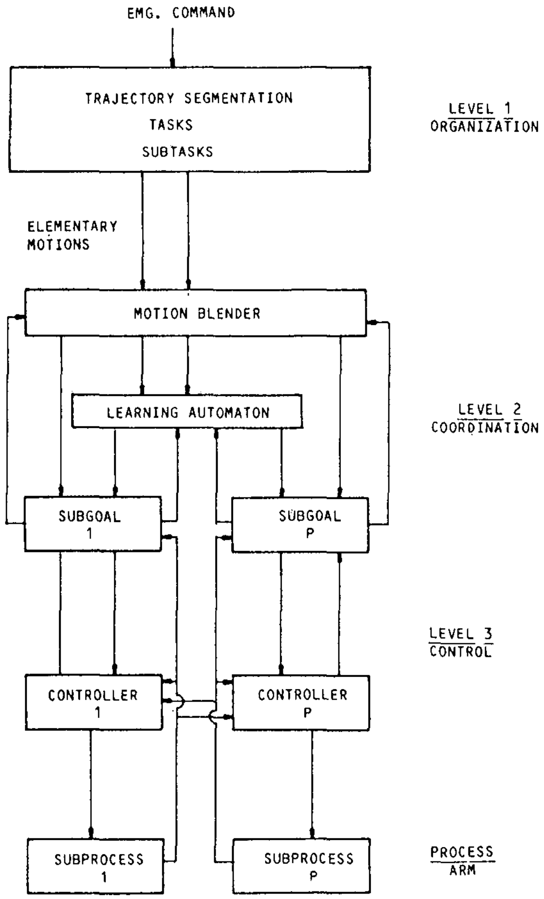
4. Conclusions

Through the above example of the control of the prosthetic arm the concept of a hierarchically intelligent man-machine interactive control system has been proposed and its feasibility established. Generalization to other man-machine interactive systems or even autonomous robots should be straightforward and would be one of the areas of future research in control systems. Intelligent control looks like the natural successor of currently researched Adaptive Learning and Self-Organizing Control Systems. Such a direction has been selected because of the impact of the digital computer in modern technology. Intelligent controls should represent the perfect matching between a lower level control hardware and a digital computer for higher level decision-making according to the principle of increasing intelligence with decreasing precision in a hierarchical control structure.

Application of intelligent controls has tremendous potential for large scale industrial processes and the development of highly sophisticated anthropomorphic robots, without excluding the possibilities of utilization in modern socioeconomic, transportation and other such systems.

Acknowledgements

This work was partially supported by NSF Grant ENG-78-02400.



Hierarchical intelligent control of a prosthetic arm.

5. References

[1] Agin, G.J., Binford, T.O., "Computer Description of Curved Objects," Proc. 3rd Int. Joint Conference on Artificial Intelligence, Stanford, Ca., Aug. 1973.

[2] Bejczy, A.K., "Remote Manipulator System Technology Review," Tech. Rep. No. 760-77, Jet Propulsion Lab., Pasadena, CA, July 1972.

[3] Bejczy, A.K., "New Techniques for Terminal Phase Control of Manipulators," Tech. Rep. 760-98, Jet Propulsion Lab., Pasadena, CA, Feb. 1974.

[4] Bejczy, A.K., "Robot Arm Dynamics and Control," Tech. Rep. 33-669, Jet Propulsion Lab., Pasadena, CA, Feb. 1970.

[5] Bejczy, A.K., "External State Sensors and Control Tutorial Workshop," session 4 at 18th IEEE Conference on Decision and Control, Ft. Lauderdale, FL, Dec. 11, 1979.

[6] Birk, J. et al, "Determining Work Piece Orientation in a Robot Hand Using Vision," Proc. 17th Conference on Decision and Control, San Diego, CA, Jan. 1979.

[7] Chandrasekaran, B. and Shen, D.W.C., "Adaptation of Stochastic Automata in Non-Stationary Environments," <u>Proc. Natl. Electron. Conf.</u>, <u>23</u>, 39-45, 1967.

[8] Feigenbaum, E.A. and Feldman, J. (eds.), <u>Computers and Thought</u>, McGraw-Hill, New York, 1963.

[9] Firschein, O., Fischier, M., Coles, S.L., and Tenenbaum, J.M., "Forecasting and Assessing the Impact of Artificial Intelligence on Society," Tech. Rep., Stanford Research Institute, Menlo Park, CA, Feb. 1973.

[10] Freedy, A., Hull, F.C., Lucaccini, L.F., and Lyman, J., "A Computer-Based Learning System for Remote Manipulator Control," <u>IEEE Trans. Systems Man Cybern.</u>, <u>SMC-1</u>, (4), 356-364, 1971.

[11] Freedy, A. and Weltman, G., "A Prototype Learning System as a Potential Controller for Industrial Robot Arms," Proc. Second International Symposium on Industrial Robots, Chicago, IL, May 16-18, 1972.

[12] Fu, K.S., "Learning Control Systems and Intelligent Control Systems: An Intersection of Artificial Intelligence and Automatic Control," <u>IEEE Trans. Automatic Control</u>, <u>AC-16</u>, (1) 70-72, 1971.

[13] Fu, K.S. (ed.), <u>Pattern Recognition and Machine Learning</u>, Plenum Press, New York, 1971.

[14] Fu, K.S., <u>Sequential Methods in Pattern Recognition and Machine Learning</u>, Academic Press, New York, 1968.

[15] Fu, K.S., Syntactic Methods in Pattern Recognition, Academic Press, New York, 1974.

[16] Fu, K.S., "Stochastic Automata as Models of Learning Systems," Computer and Information Sciences-II, ed., J.T. Tou, Academic Press, 1967.

[17] Fu, K.S., "On Syntactic Pattern Recognition and Stochastic Languages," in Frontiers of Pattern Recognition (S. Watanabe, ed.), Academic Press, New York, 1972.

[18] Fu, K.S., "Learning Techniques in System Design -- A Brief Review," Fifth World Congress of IFAC, Paris, June 12-16, 1972.

[19] Fu, K.S. and Fung, L.W., "Decision-Making in a Fuzzy Environment," Tech. Rept. TR-EE 73-22, Purdue University, West Lafayette, IN, May 1973.

[20] Fu, K.S., and McLaren, R.W., "An Application of Stochastic Automata to the Synthesis of Learning Systems," Tech. Rept. TR-EE 65-17, Purdue University, West Lafayette, IN, Sept. 1965.

[21] Fu, K.S. and Nikolic, Z.J., "A Study of Learning Systems Operating in Unknown Stationary Environments," Tech. Rept. TR-EE 66-20, Purdue University, West Lafayette, IN, Nov. 1966.

[22] Graupe, D., "Control of Artificial Upper Extremity Prostheses in Several Degress of Freedom," Progress Report VA Grant V10(134), Dept. of Electrical Engineering, Colorado State University, Fort Collins, CO, Feb. 1974.

[23] Graupe, D. et al, "A Microprocessor System for Multifunctional Control of Upper Limb Prostheses via EMG Identification," IEEE Trans. on Aut. Control Vol. AC-23, pp. 538-544, Aug. 1978.

[24] Karchak, A., Jr. and Allen, J.B., "Investigation of Powered Orthotic Devices," Final Report VRA Grant RD-1461-M-67, Rancho Los Amigos Hospital, University of Los Angeles, CA, 1967.

[25] Klinger, A., "Natural Language, Linguistic Processing, and Speed Understanding: Recent Research and Future Goals," UCLA Tech. Report R-1377ARPA Rept., Los Angeles, CA, Dec. 1973.

[25] Lyman, J. and Freedy, A., "Summary of Research Activities on Upper Prosthesis Control," Bull. Prosthetic Res., Winter, 1973.

[26] McGee, R.B., Control of Legged Locomotion Systems, Proceedings 1977 JACC, San Francisco, June 1977.

[27] Mendel, J.M. and Fu, K.S. (eds.), Adaptive, Learning, and Pattern Recognition Systems Theory and Applications, Academic Press, New York, 1970.

[28] Minsky, M.L., Artificial Intelligence, McGraw-Hill, New York, 1972.

[29] Minsky, M.L., "Artificial Intelligence and Intelligent Automata," Project MAC Progress Report, 1968-1969, MIT, Cambridge, MA, 1969.

[30] Minsky, M.L. and Papert, S.A., "Research on Intelligent Automata," Project MAC, Status Report II, MIT, Cambridge, MA, Sept. 1967.

[31] Nikolic, Z.J. and Fu, K.S., "An Algorithm for Learning Without External Supervision and Its Application to Learning Control Systems," IEEE Trans. on Auto. Control, Vol. AC-11, July 1966.

[32] Nilsson, N.J., "A Mobile Automation: An Application of Artificial Intelligence Techniques," Proc. International Joint Conference on Artificial Intelligence, Washington, DC, May 1969.

[33] Nilsson, N.J., "Current Artificial Intelligence Research at SRI," Oral Presentation, University of California, Los Angeles, May 1971.

[34] Nilsson, N.J., Learning Machines, McGraw-Hill, New York, 1965.

[35] Norman, M.F., "Mathematical Learning Theory," in Mathematics of the Decision Sciences, (G.B. Dantzig and A.F. Veinott, eds.), Am. Math. Soc. Publications, New York, 1968.

[36] Paul, R.P., "Kinematic Control Equations for Simple Manipulators," Proc. 17th Conference on Decision & Control, San Diego, CA, Jan. 1979.

[37] Rosen, C.A. and Nilsson, N.J., "An Intelligent Automation," 1967 IEEE International Convention Record, Part 9, New York, March 1967.

[38] Saridis, G.N., Self-Organizing Control of Stochastic Systems, Marcel Dekker, New York, 1977.

[39] Saridis, G.N., "Self-Organizing Control and Application to Trainable Manipulators and Learning Prostheses," Proc. Sixth IFAC Congress, Cambridge, MA, Aug. 1975.

[40] Saridis, G.N., "Toward the Realization of Intelligent Controls," IEEE Proceedings, Vol. 67, No. 8, Aug. 1979.

[41] Saridis, G.N., Graham, J., Lee, G., "An Integrated Syntactic Approach and Suboptimal Control for Manipulators and Prostheses," Proc. of 18th Conference on Decision and Control, Ft. Lauderdale, FL, Dec. 1979.

[42] Saridis, G.N. and Lee, C.S.G., "Computer Controlled Manipulators with Visual Inputs," Optical Engineering, Vol. 18, No. 5, Sept. 1979.

[43] Saridis, G.N. and Lee, C.S.G., "On Hierarchically Intelligent Control and Management of Traffic Systems," Computer Control of Urban Systems, ASCE Publication, New York, 1979.

[44] Saridis, G.N. and Lee, C.S.G., "Heuristic Control of Trainable Manipulators," Proceedings 1976 JACC, pp. 712-716, W. Lafayette, IN, July 1976.

[45] Saridis, G.N. and Lee, C.S.G., "Approximation Theory of Optimal Control for Trainable Manipulators," IEEE Transactions on S.M.C., Vol. SMC-8, No. 3, March 1979.

[46] Saridis, G.N. and Newman, M.A., "Upper Limb EMG Statistical Analysis," MIDCON '79, Chicago, IL, Nov. 1979.

[47] Saridis, G.N. and Stephanou, H.E., "A Hierarchical Approach to the Control of a Prosthetic Arm," IEEE Transactions on SMC, Vol. SMC-7, No. 6, pp. 407-420, June 1977.

[48] Shapiro, I.J. and Narendra, K.S., "Use of Stochastic Automata for Parameter Self-Optimization with Multimodal Performance Criteria," IEEE Trans. on System Science and Cybernetics, Vol. SSC-5, Oct. 1969, and Witten, I.H., "Comments on 'Use of Stochastic Automata for Parameter Self-Optimization with Multimodal Performance Criteria," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-2, April 1972.

[49] Sklansky, J., "Learning Systems for Automatic Control," IEEE Trans. Automatic Control, (1) 6-20, 1966.

[50] Slagle, J.R., Artificial Intelligence; The Heuristic Programming Approach, McGraw-Hill, New York, 1971.

[51] Tsypkin, Ya. Z., "Generalized Learning Algorithms," Automation and Remote Control, (1), 86-92, 1970.

[52] Whitney, D.E., "Resolved Motion Control of Manipulators and Human Prosthesis," IEEE Trans. Man-Machine Systems, MMS-10, (2), 1969.

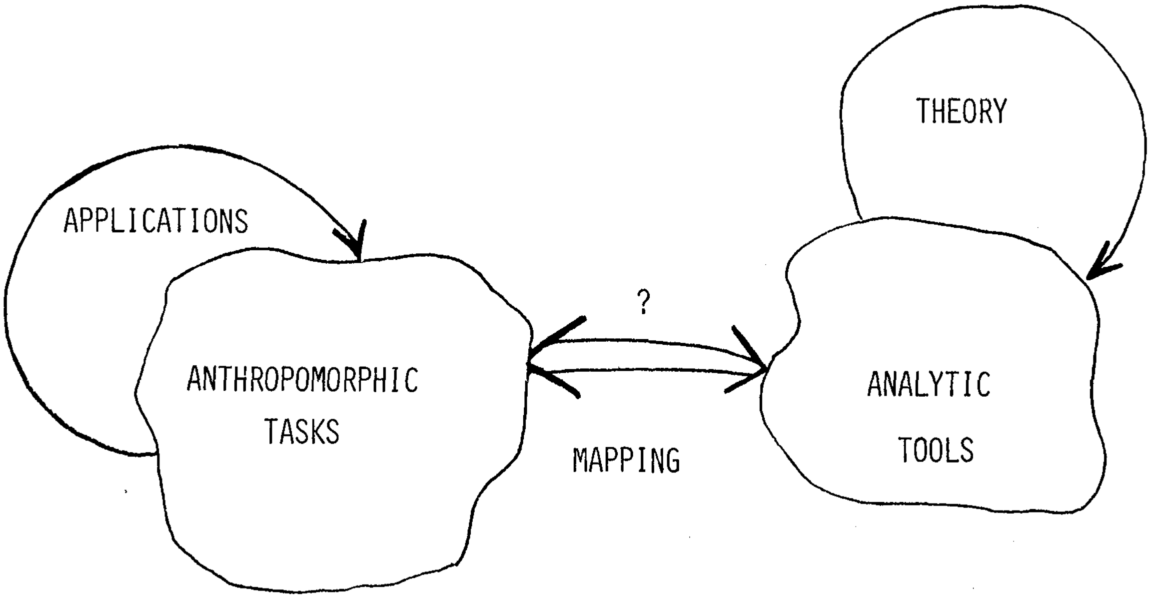
[53] Whitney, D.E., "State Space Models of Remote Manipulation Tasks," Proc. 1969 International Joint Conference on Artificial Intelligence, Washington, DC, 1969.

[54] Widrow, B., "Generalization and Information Storage in Networks of Adaline 'Neurons'," in Self-Organizing Systems (M.C. Yovits, G.T. Jacobi, and C.D. Goldstein, eds.), Spartan Books, Washington, DC, 1962.

[55] Winston, P.E., "Learning Structural Descriptions from Examples," Tech. Rept. Project MAC TR-76, MIT, Cambridge, MA, 1970.

[56] Zadeh, L.A., "Fuzzy Sets," Inform. Control, 8, (3), 338-355, 1965.

This is a flowchart which represents the function of intelligent machines as a mapping between anthropomorphic tasks and analytic operations.



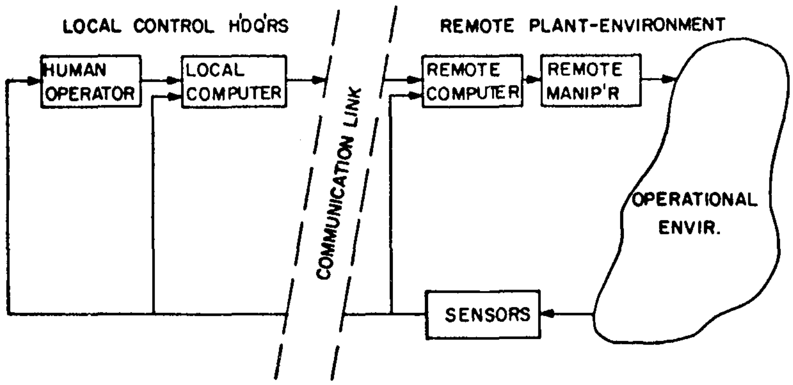
- COMMANDS
- FEEDBACK
- INFORMATION EXCHANGE

- COGNITIVE ENGINEERING
- SYSTEMS ENGINEERING
- COMPUTER ENGINEERING
- INTEGRATION

THE PROBLEM OF DESIGNING MACHINES TO

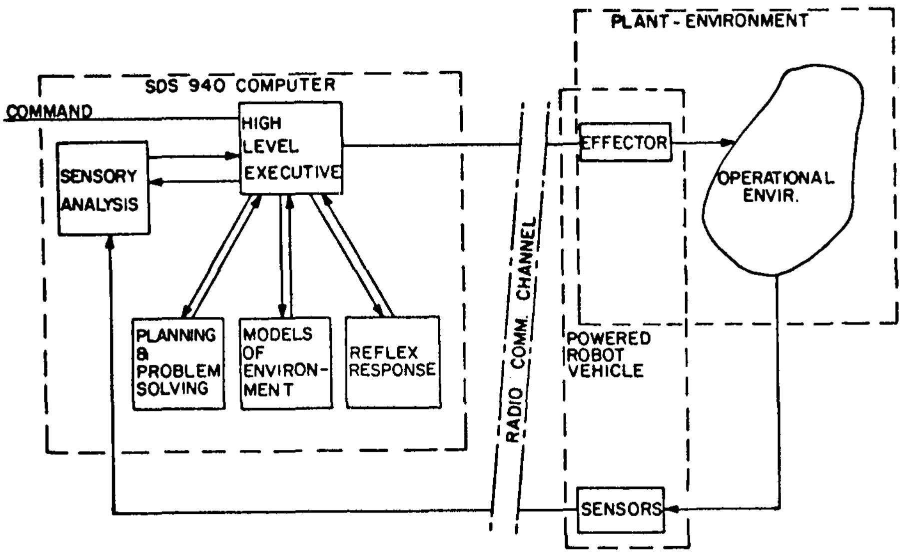
EXECUTE ANTHROPOMORPHIC FUNCTIONS

This block diagram outlines the basic functions of a man-machine interactive manipulator.



Man-machine interactive remote-controlled manipulator (example of servo-operated system).

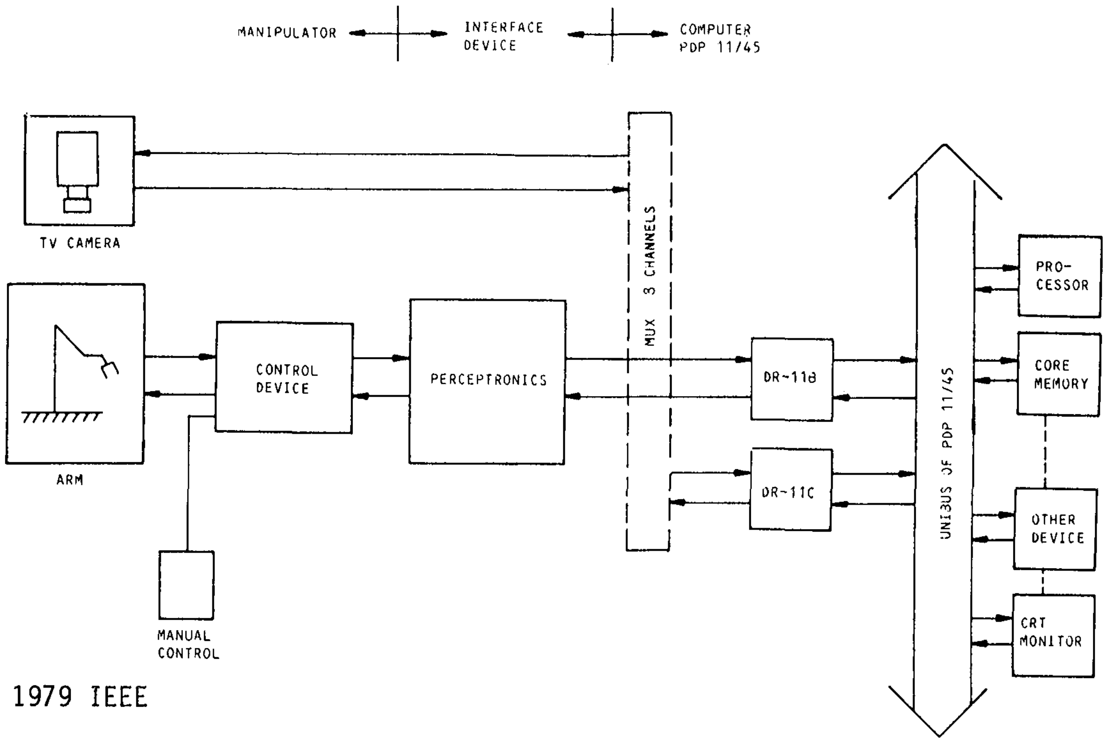
This is a block diagram of the functions of the SRI robot Shakey.



The SRI robot system "Shakey" (example of "problem solving approach" operated system).

Reprinted from reference 38, pp. 454 and 457, by courtesy of Marcel Dekker, Inc.

This is a block diagram showing the components of the "intelligent arm" of the Advanced Automations Research Laboratory at Purdue University.



"INTELLIGENT CONTROL"

A

HIERARCHICAL MARRIAGE

OF

COMPUTER SOFTWARE

AND

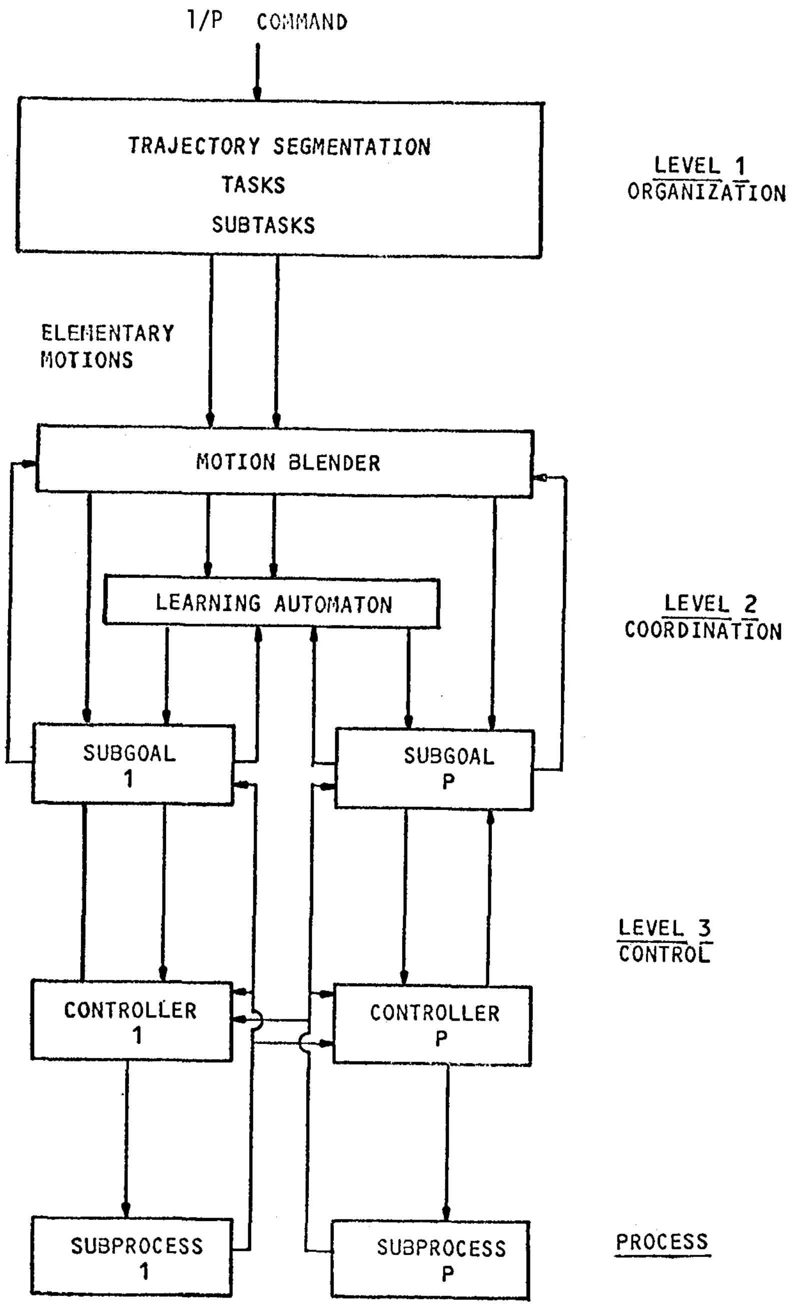
CONTROL SYSTEMS

OR A

MANAGERIAL APPROACH

TO PROCESS CONTROL

This is a block diagram of the structure of a "hierarchically intelligent control" system.



THE HIERARCHICALLY INTELLIGENT CONTROL STRUCTURE c. 1979 IEEE

This gives a theoretical justification of "hierarchically intelligent control."

THEORETICAL APPROACH:

DECENTRALIZATION OF CONTROL PROCESS, HIERARCHICAL APPROACH, MULTIPLE PERFORMANCE CRITERIA, SYNTHESIS OF ANALYTICAL SYSTEM THEORETIC METHODS AND HEURISTIC ARTIFICIAL INTELLIGENCE TECHNIQUES.

THREE HIERARCHICAL ASPECTS:

- VERTICAL DECOMPOSITION ACCORDING TO FUNCTION (LEVELS)
- HORIZONTAL DECOMPOSITION ACCORDING TO SUBPROCESS (STRATA)
- SEQUENTIAL SOLUTION, TEMPORAL DECOMPOSITION (LAYERS)

EXISTING METHODS: ONE GOAL, ONE LEVEL, ONE STRATUM, ONE LAYER, LINEARIZATION ABOUT NOMINAL TRAJECTORY.

CONTRIBUTIONS:

- HIERARCHICAL DECOMPOSITION
- NONLINEAR PERFORMANCE ADAPTIVE CONTROL STRUCTURE
- LINGUISTIC COORDINATION, APPLICATION OF FORMAL LANGUAGES TO DYNAMIC SYSTEMS

PRINCIPLE OF INCREASING INTELLIGENCE AND DECREASING PRECISION.

This is the formulation of hierarchically intelligent control for a prosthetic arm.

PROBLEM FORMULATION

7-DEGREE-OF-FREEDOM PROSTHESIS, 7 DC MOTORS WITH POTENTIOMETERS AND TACHOMETERS, COMPUTER CONTROL.

EXISTING DEVICES: · MUSCULAR CONTROL BY BOWDEN CABLES, REQUIRE CONSIDERABLE MENTAL EFFORT, SLOW.

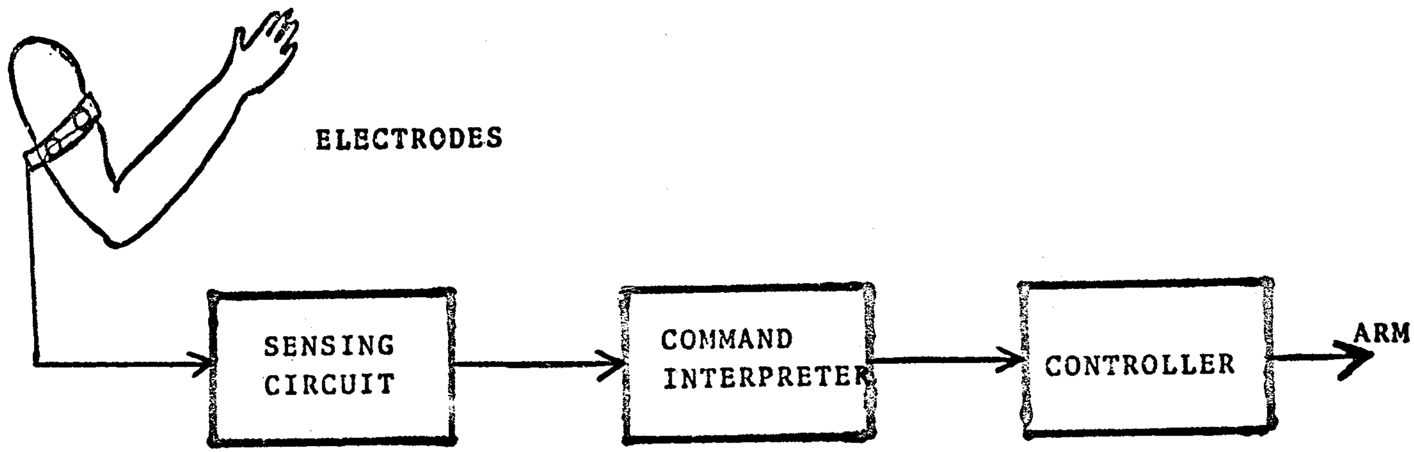
MAIN OBJECTIVE: AUTOMATE PART OF THE DECISION-MAKING PROCESS BY ESTABLISHING COMPUTER REFLEXES WITH LEARNING CAPABILITY.

DEFICIENT COMMUNICATION WITH THE CENTRAL NERVOUS SYSTEM, REPLACE 52 MUSCLE PAIRS BY 7 MOTORS.

PERFORMANCE CRITERIA: ANTHROPOMORPHIC TRAJECTORIES, LOW ACCELERATION, LOW ENERGY CONSUMPTION.

MOTION OF THE WRIST, ORIENTATION OF THE HAND.

Here is shown the EMG Signal Generation for the hierarchically intelligent control of a prosthetic arm.



The EMG encoding and decoding and control system.

Here we have the binary code used for EMG signal command and its linguistic interpretation.

Command Vocabulary

Binary Code	Command Primitive	Terminal Symbol
0001	1	T_1
0010	2	T_2
0011	3	T_3
0100	4	T_4
0101	5	T_5
0110	6	T_6
0111	7	T_7
1000	8	T_8
1001	Go to	T_9
1010	Move joint	T_{10}
1011	Orient hand	T_{11}
1100	Not	T_{12}
1101	Very	T_{13}
1110	Fast	T_{14}
1111	Stop	T_{15}

This is an explanation of the linguistic organization level.

ORGANIZATION LEVEL

ELECTROMYOGRAPHIC SIGNALS, PULSE WIDTH MODULATION, BINARY CODING, INVARIANCE TO NOISE, FATIGUE, MUSCULAR WEAKNESS, EASY TO LEARN, MORSE CODE.

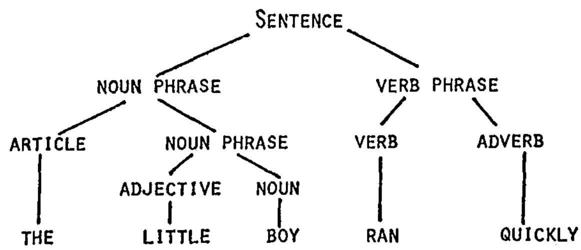
15-WORD VOCABULARY, 4-BIT WORDS, LINGUISTIC STRINGS, OR SENTENCES.

RECOGNITION OF SYNTACTICALLY VALID COMMANDS, MAPPING INTO SEQUENCE OF STATE SPACE GOALS, I.E. DESIRED ANGULAR POSITIONS.

MOTION REDUNDANCY: FOUR ANGULAR DEGREES OF FREEDOM, ONLY THREE CARTESIAN COORDINATES, NONUNIQUE TRANSFORMATION, LEARNING, LOW ENERGY, SMALL ANGULAR DISPLACEMENTS, ANTHROPOMORPHIC MOTION.

This is an example of formal grammar and languages.

INTRODUCTION TO GRAMMARS



$S \to NP.\ VP$	(1)
$NP \to A.\ NP$	(2)
$VP \to V.\ ADV$	(3)
$NP \to ADJ.\ N$	(4)
$A \to$ THE	(5)
$ADJ \to$ LITTLE	(6)
$N \to$ BOY	(7)
$V \to$ RAN	(8)
$ADV \to$ QUICKLY	(9)

FOUR CONCEPTS:

NONTERMINALS: S, NP, VP, N, ADJ, ADV, A, V
TERMINALS: THE, LITTLE, BOY, RAN, QUICKLY
START SYMBOL: S
REWRITING, OR PRODUCTION RULES (1)...(9)

This is an example of a parsing algorithm for formal languages and advantages of the Cocke-Younger-Kasami parser which was used in the prosthetic arm.

A PARSING ALGORITHM

FOUR TYPES OF GRAMMARS

1. UNRESTRICTED
2. CONTEXT SENSITIVE: $\omega_1 A \omega_2 \to \omega_1 \beta \omega_2$
3. CONTEXT FREE: $A \to \beta$
4. FINITE STATE: $A \to aB,\ A \to a$

CHOMSKY NORMAL FORM: $A \to BC$

$A \to a$

COCKE-YOUNGER-KASAMI PARSING ALGORITHM:

- DYNAMIC PROGRAMMING CONCEPT
- EASY TO IMPLEMENT
- ADAPTABLE TO STOCHASTIC AND FUZZY GRAMMARS WITH ON-LINE LEARNING
- TIME $\sim N^3$, SPACE $\sim N^2$

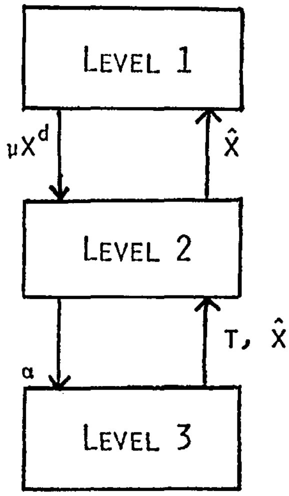
These are the syntactic rules used for command generation of the prosthetic arm.

Command Syntax Rules

1. $N_1 \to T_{15}$	13. $N_5 \to T_3$
2. $N_1 \to N_4N_3$	14. $N_5 \to T_4$
3. $N_2 \to T_{12}$	15. $N_5 \to T_5$
4. $N_2 \to T_{13}$	16. $N_5 \to T_6$
5. $N_3 \to N_2N_3$	17. $N_5 \to T_7$
6. $N_3 \to T_{14}$	18. $N_5 \to T_8$
7. $N_4 \to N_4N_{10}$	19. $N_6 \to T_9$
8. $N_4 \to N_6N_7$	20. $N_7 \to N_5N_8$
9. $N_4 \to N_9N_8$	21. $N_8 \to N_5N_5$
10. $N_5 \to N_2N_5$	22. $N_9 \to T_{10}$
11. $N_5 \to T_1$	23. $N_{10} \to N_{11}N_8$
12. $N_5 \to T_2$	24. $N_{11} \to T_{11}$

Here we have the formulation of the coordination problem as a minimum energy problem and a block diagram of the linguistic parser to implement the first part of the coordinating controller.

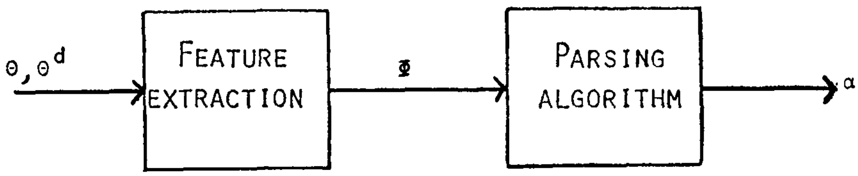
FORMULATION OF THE COORDINATION PROBLEM



MINIMUM ENERGY

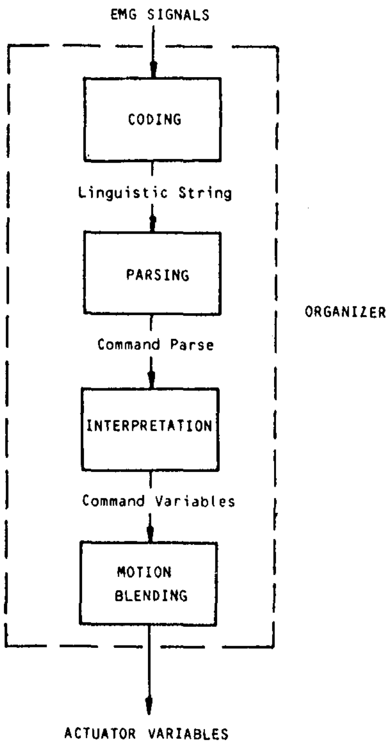
$$_i(\alpha) = \lim_{\tau \to \infty} \frac{1}{\tau} E \int_0^{\tau} |\dot{\theta}_i(t)\ T(t)|\, dt$$

$$J(\alpha, \mu) = \sum_{i=1}^{4} \mu_i\ J_i(\alpha)$$



$$\Theta = [\theta_1, \theta_2, \theta_3, \theta_4]^T$$

This block diagram shows all the functions of the organization level.



The organization level of an "intelligent" prosthetic arm.

COORDINATION LEVEL

FOR GIVEN CONTROL PATTERN (INITIAL, FINAL POSITION, VELOCITY), SELECT COORDINATION INPUT, SYNERGY OF CONTROLLERS

OVERALL DYNAMIC PERFORMANCE CRITERION, MINIMUM ENERGY.

FEATURE EXTRACTION

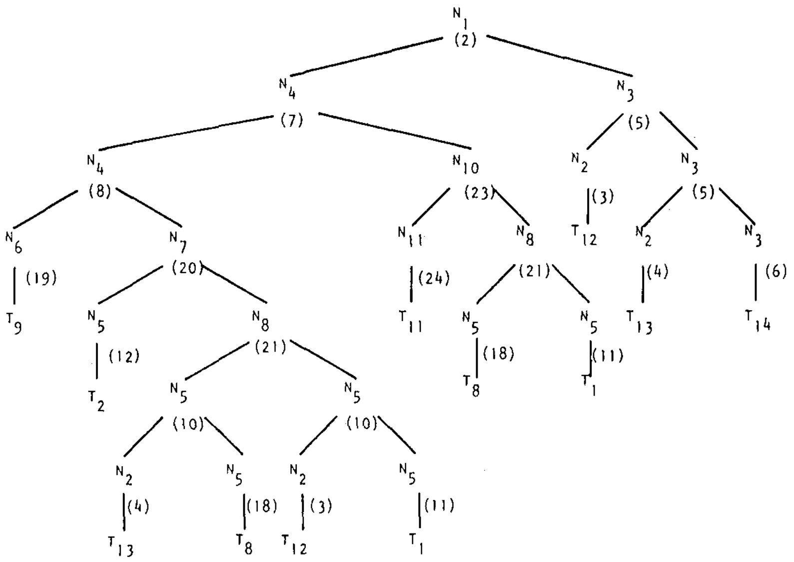
PARTITION OF FEATURE SPACE BY CONTEXT FREE GRAMMARS IN CHOMSKY NORMAL FORM.

SYNTACTIC CLASSIFICATION INTO COORDINATION CLASSES.

MULTILAYERED PARSING ALGORITHM OF DYNAMIC PROGRAMMING TYPE.

LEARNING BY GRAMMATICAL INFERENCE

This is a command parse example for the prosthetic arm.



String: T_9 T_2 T_{13} T_8 T_{12} T_1 T_{11} T_8 T_1 T_{12} T_{13} T_{14}

(go to) (2) (very 8) (not 1) (orient hand) (8) (1) (not) (very) (fast)

c. 1977 IEEE

These are the command executions at the organization level.

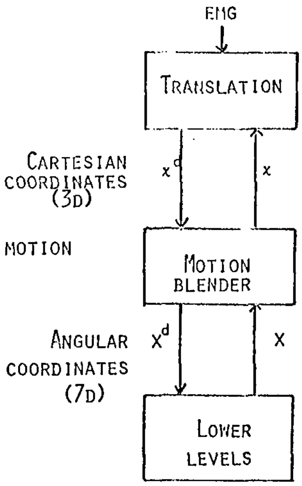
MAN-MACHINE INTERFACE

LEVEL 1: ORGANIZATION

EXECUTIVE LEVEL
PURPOSIVE COMMANDS (TASK ORIENTED)
INPUT: LINGUISTIC STRING
OUTPUT: SEQUENCE OF GOALS IN STATE SPACE (MACHINE LANGUAGE)
MOTION BLENDER:
i) REDUCTION OF REDUNDANCY IN WRIST MOTION
ii) HAND ORIENTATION

TERMINAL SYMBOLS:

GO TO	(X, Y, Z)
MOVE JOINT	(I, θ)
SET SPEED	(FAST SLOW)
AVOID	(OBJECT)
FOLLOW	(CONTOUR)
CONSTRAINT H	(OBJECT IN HAND HORIZONTAL)
CONSTRAINT V	(OBJECT IN HAND VERTICAL)

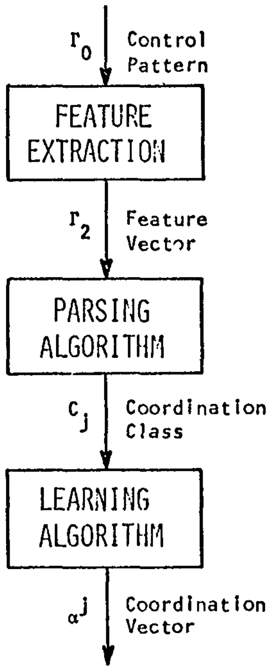


This is an introduction to fuzzy set theory with comparison with probabilistic models.

FUZZY SETS

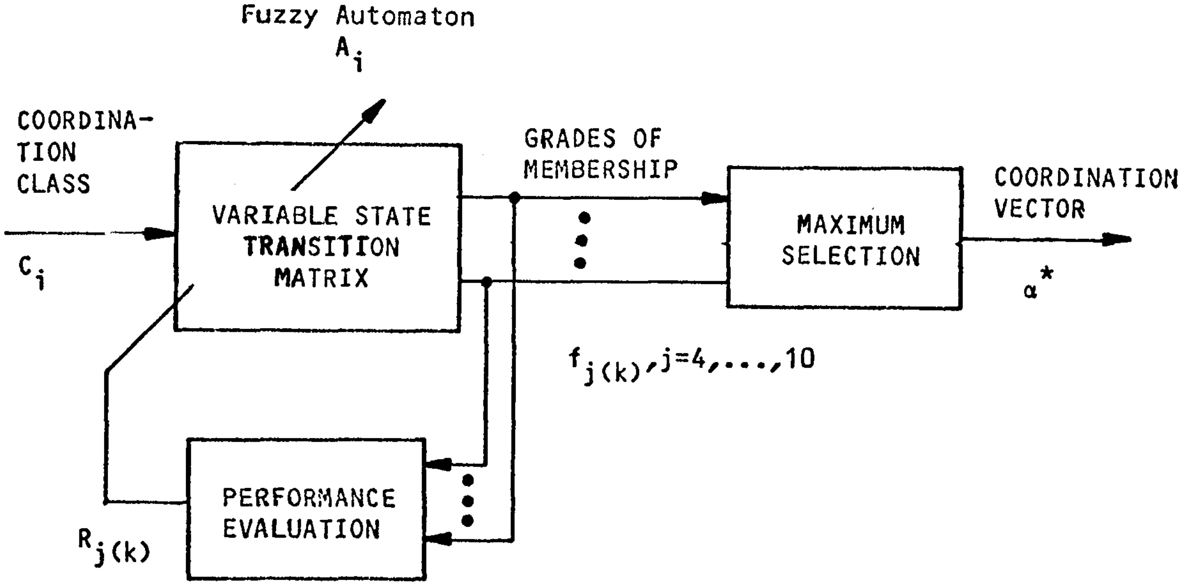
FUZZY CONCEPT	PROBABILISTIC ANALOG
$A = (x, \mu A(x))$	$A = (x, PA(x))$
$\mu A(x)$: GRADE OF MEMBERSHIP OF IN A	$PA(x)$: PROBABILITY THAT BELONGS TO A
$\mu A\cap B(x) = \text{MIN}\ (\mu A(x), \mu B(x))$	$PA\cap B(x) = PA(x) \cdot PB(x)$
$\mu A\cup B(x) = \text{MAX}\ (\mu A(x), \mu B(x))$	$PA\cup B(x) = PA(x) + PB(x)$
$\mu B(y) = \text{SUP}_x\ \text{MIN}\ (\mu A(x), \mu B(y/x))$	$PB(y) = \sum_x PA(x) \cdot PB(y/x)$

Block diagram of the coordinator showing cascaded the linguistic parser and the "fuzzy automaton" which provides the learning algorithms.



Linguistic Coordination Block Diagram

This shows the "fuzzy automaton" as an on-line learning scheme for the coordination of motion of the prosthetic arm.



c. 1977 IEEE

CONTROL LEVEL

MOTION OF THE WRIST, 4 DEGREES OF FREEDOM, BIOLOGICAL PRINCIPLE OF LEAST INTERACTION, EACH JOINT TRAJECTORY AS IF SUBSYSTEM ISOLATED FROM ENVIRONMENT AND COUPLING EFFECTS.

CONTROL STRUCTURE: LINEAR PART + NONLINEAR PART, CONTROL VARIABLE OBTAINED BY NONLINEAR TRANSFORMATION APPLIED TO ACTUAL INPUT, I.E. TORQUE.

PARTIAL KNOWLEDGE OF DYNAMICS, COMPUTATIONAL SIMPLICITY, UNKNOWN PARAMETERS, PERFORMANCE ADAPTIVE APPROACH.

EXPANDING SUBINTERVAL ALGORITHM FOR ON-LINE LEARNING, REDUCTION OF UNCERTAINTIES AS PROCESS EVOLVES, KINEMATIC PERFORMANCE CRITERION.

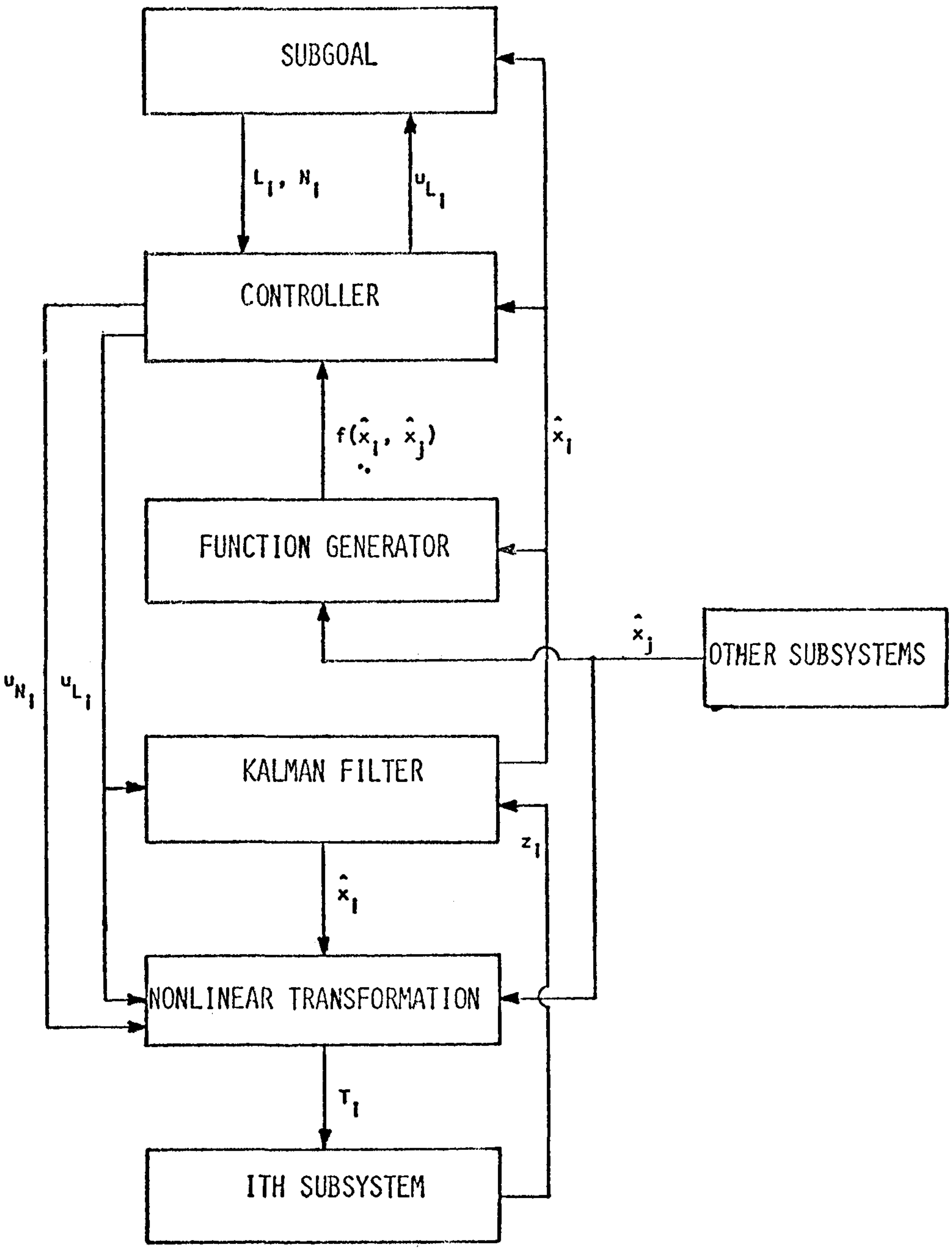
MULTILAYERED LEARNING TECHNIQUE FOR ACCELERATION OF CONVERGENCE RATE, KALMAN FILTERING FOR STATE ESTIMATION.

EACH LOCAL PERFORMANCE CRITERION HAS A VARIABLE PARAMETER, OR COORDINATION INPUT, SELECTED BY HIGHER LEVEL.

DIRECT INPUT TO MOTORS

REFLEX ACTION

This is the block diagram of the self-organizing adaptive hardware controller for the i-th subsystems.



Structure of the nonlinear performance adaptive controller

DYNAMIC MODEL OF THE ARM FOR TWO-DIMENSIONAL MOTION:

SHOULDER AND ELBOW FLEXION-EXTENSION

$$\begin{bmatrix} \ddot{\theta}_2 \\ \ddot{\theta}_4 \end{bmatrix} = \frac{1}{c_1 + c_2 \sin^2\theta_4} \begin{bmatrix} c_3 & c_4 + c_5 \cos\theta_4 \\ c_6 + c_7 \cos\theta_4 & c_8 + c_9 \cos\theta_4 \end{bmatrix}$$

$$\begin{bmatrix} c_{10}\dot{\theta}_4(\dot{\theta}_4 + 2\dot{\theta}_2)\sin\theta_4 + c_{11} \cos\theta_2 + c_{12} \cos(\theta_2 + \theta_4) + T_2 \\ c_{13}\dot{\theta}_2\dot{\theta}_2 \sin\theta_4 + c_{14}\theta_2(\theta_2 + \theta_4)\sin\theta_4 + c_{15} \cos(\theta_2 + \theta_4) + T_4 \end{bmatrix}$$

$$\ddot{\theta} = D(\theta,\dot{\theta})\ [H(\theta,\dot{\theta}) + T]$$

$$= F(\theta,\dot{\theta}) + U$$

$$F(\cdot) = D(\cdot)H(\cdot)$$

$$U = D(\cdot)T$$

FORMULATION OF THE DIRECT CONTROL PROBLEM FOR ONE SUBSYSTEM

$$\ddot{\theta} = f(\theta,\dot{\theta}) + u + n$$

$$\dot{x} = Ax + B\ [f(x,\ y) + u + n]$$

PLANT MEASUREMENT

$$z = Cx + v$$

$$X = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

PERFORMANCE CRITERION:

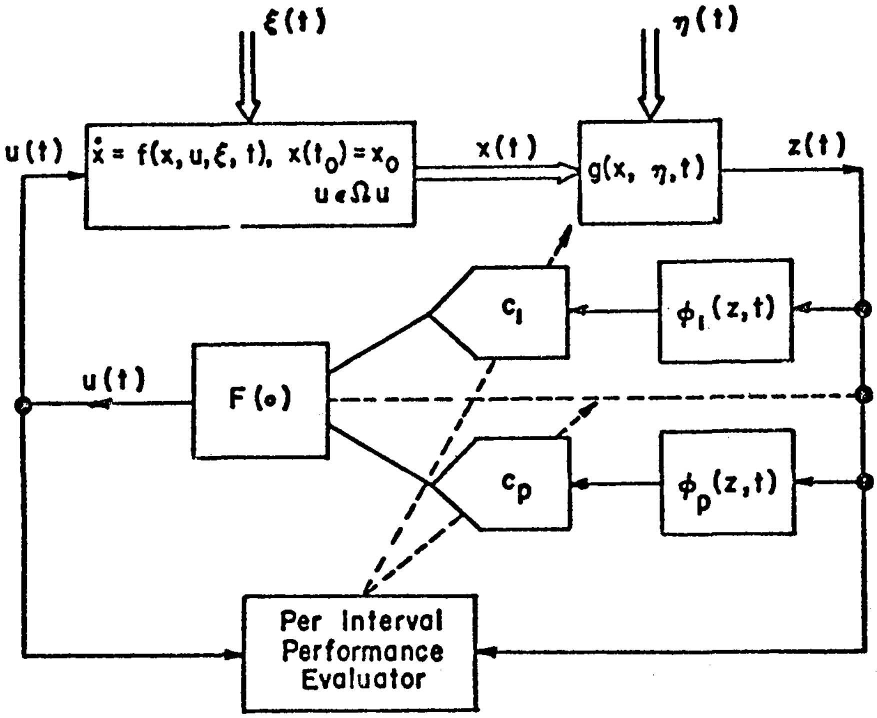
$$J_1(\tau) = \lim_{\tau\to\infty} \frac{1}{2} E \int_Q^{\tau} (||\hat{x}(t) - \hat{x}^d(\tau)||^2 Q(\alpha) + u_L(t)^2)dt \qquad J = \sum_{i=1}^{4} \mu_i J_i(\alpha)$$

CONTROLLER STRUCTURE:

$$u = L(\hat{x} - x^d) + K\ f(\hat{x},\ \hat{y})$$

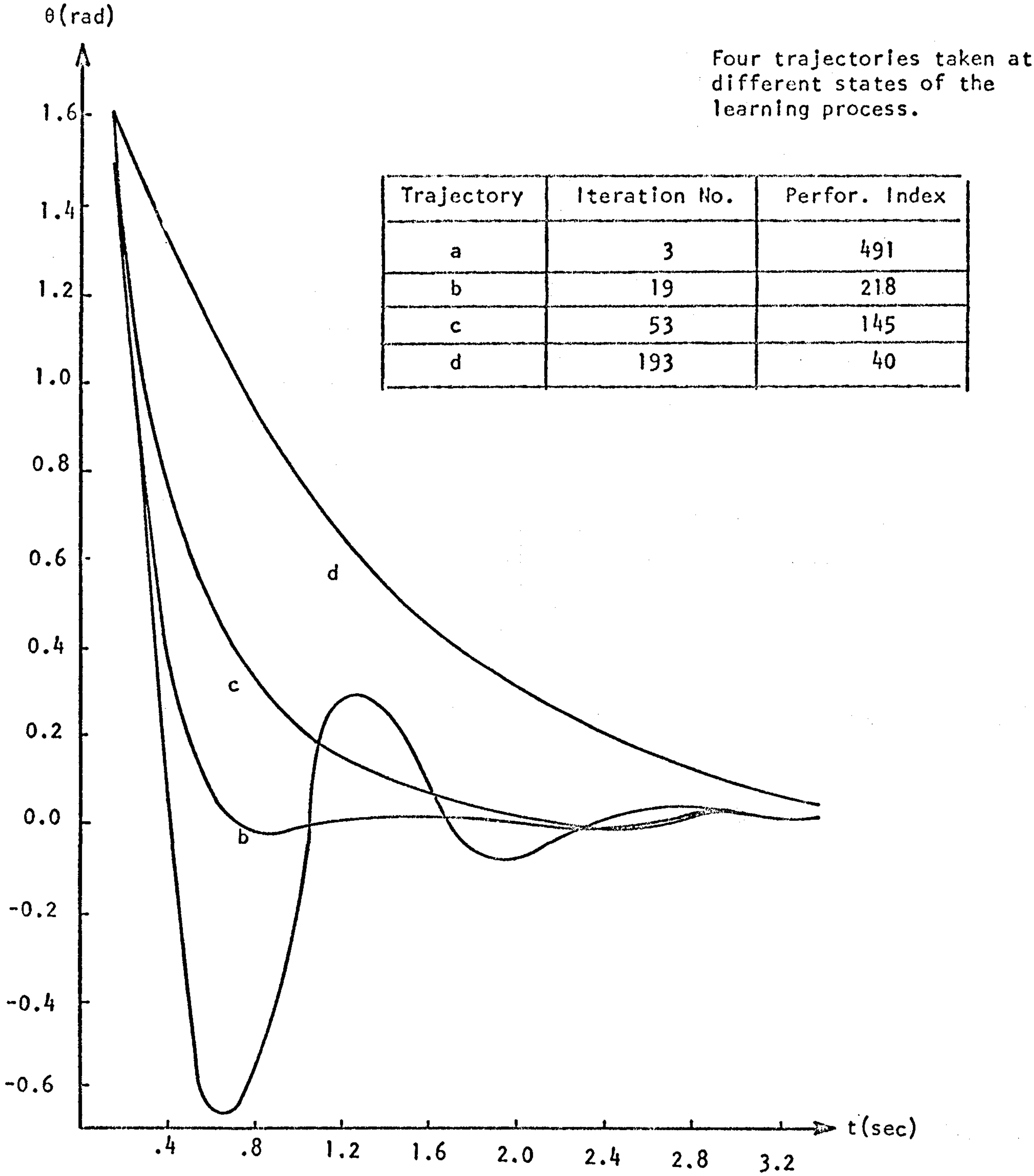
$$= u_L + u_K$$

This is a general block diagram of a self-organizing performance adaption control providing on-line learning for the improvement of its performance.



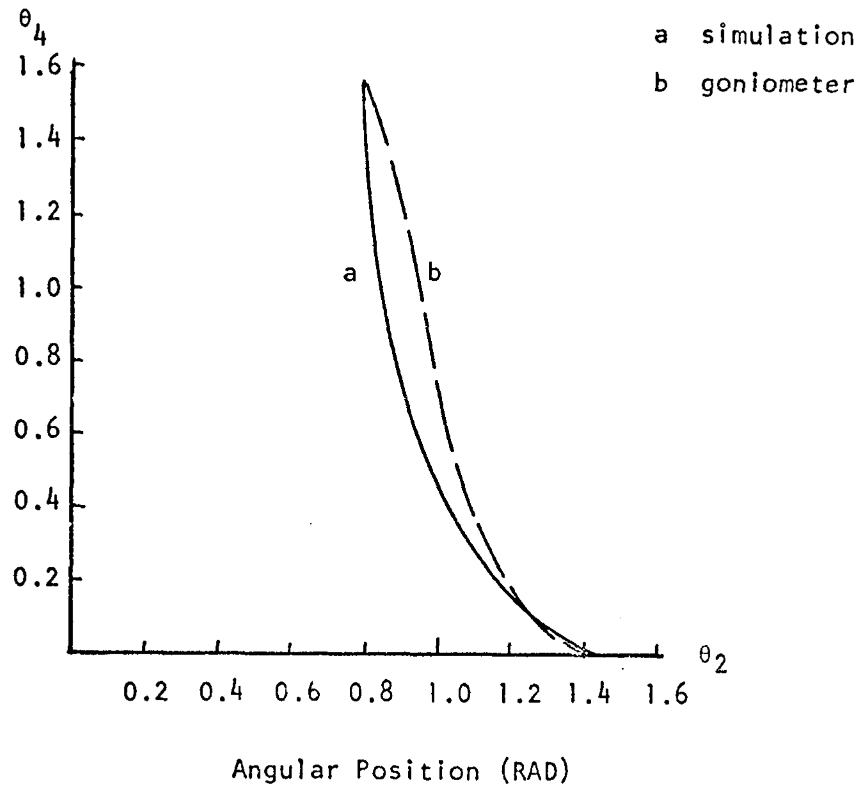
THE SELF-ORGANIZING PERFORMANCE ADAPTIVE CONTROL SYSTEM OF SARIDIS

These are simulation results of the application of self organizing performance adaptive algorithm showing the effects of learning after several iterations (curve d).



Trajectory	Iteration No.	Perfor. Index
a	3	491
b	19	218
c	53	145
d	193	40

This is a comparison of simulation results of a complete coordinated motion after a period of learning, to the actual motion of a human arm recorded through a "goniometer."



Phase Plane Trajectory for a Coordinated Motion

DECENTRALIZED STOCHASTIC CONTROL

JASON L. SPEYER
UNIVERSITY OF TEXAS

The objective is to present a perspective of the state of control theory as it relates to decentralized stochastic control. Decentralized stochastic control is characterized by being:

1. Decentralized in that the information to one controller is not the same as information to another controller.

2. The system including the information has a stochastic or uncertain component. This complicates the development of decision rules which one determines under the assumption that the system is deterministic.

3. The system is dynamic which means the present decisions affect future system responses and the information in the system. This circumstance presents a complex problem where tools like dynamic programming are no longer applicable.

It is our intent to discuss these difficulties from an intuitive viewpoint. Particular assumptions are introduced which allow a limited theory which produces mechanizable affine decision rules.

This slide gives motivation for having a decentralized control system.

DECENTRALIZED STOCHASTIC CONTROL

APPLICATIONS

- POWER SYSTEMS
- TRANSPORTATION SYSTEMS
- AEROSPACE SYSTEMS

  AIRCRAFT (CCV), C^3 SYSTEMS

CHARACTERIZATION OF THESE SYSTEMS

- SYSTEM FORMS A NETWORK OF NODES USUALLY SPACIALLY DISTRIBUTED
- AT EACH NODE, THERE IS A CONTROLLER OR DECISION MAKER

THE OBJECTIVE OF THE CONTROL DESIGN IS THAT THE SYSTEM DEGRADE GRACEFULLY UNDER STRUCTURAL PERTURBATIONS. NETWORK STRUCTURE IMPOSED.

DECENTRALIZED CONTROL THEORY GIVES INSIGHT INTO THE STRUCTURE OF THE CONTROLLER.

This slide presents concepts associated with centralized stochastic feedback controllers. The issue is what constitutes the information necessary to construct a centralized causal stochastic controller.

CENTRALIZED STOCHASTIC CONTROL

- PROPERTIES OF THE STATE OF THE SYSTEM:
  - VALUE OF PRESENT STATE IS THE MINIMAL INFORMATION FOR PREDICTING THE FUTURE STATE
  - VALUE OF PRESENT STATE SUFFICIENT FOR CONTROLLING SYSTEM TO ACHIEVE DESIRED GOALS.

- IN STOCHASTIC SYSTEMS, THE STATE MIGHT BE THE PROBABILITY DENSITY FUNCTION OF THE STATE CONDITIONED ON THE OBSERVATION HISTORY.

- SEPARABLE CONTROLLER: THE OPTIMAL CONTROL DEPENDS UPON THE CONDITIONAL PROBABILITY DENSITY FUNCTION.

- CENTRALIZED CONTROLLERS ARE CHARACTERIZED AS SEPARABLE CONTROLLERS SINCE THERE IS ONLY ONE DECISION MAKER WITH ACCESS TO ALL INFORMATION.

  - THIS CHARACTERIZATION ALLOWS NESTING OF THE <u>CONDITIONAL EXPECTATIONS</u> USED IN THE DYNAMIC PROGRAMMING ALGORITHM.

This slide indicates the difficulties of extending centralized concepts to the decentralized problem.

DECENTRALIZED STOCHASTIC CONTROL

- MANY DECISION MAKERS.

- EACH DECISION OR CONTROL ACTION AFFECTS FUTURE DECISION OF ALL CONTROLLERS.

- CONTROL ACTION AND INFORMATION OF EACH DECISION MAKER ARE ONLY PARTIALLY SHARED.

- DETERMINATION OF A "STATE" FOR THE DECENTRALIZED PROBLEM IS NOT CLEAR IN THAT THE SUMMARY OF DATA FOR ONE CONTROLLER IS DIFFERENT FROM THE SUMMARY OF DATA FOR ANY OTHER CONTROLLER:

    - NESTING OF CONDITIONAL EXPECTATIONS IN GENERAL NOT POSSIBLE; DYNAMIC PROGRAMMING NOT APPLICABLE.

- INFORMATION STRUCTURE ALLOWING THE NESTING OF CONDITIONAL EXPECTATIONS IS CALLED A <u>CLASSICAL INFORMATION PATTERN</u>.

This presents a description of the simple problem constructed by Witsenhausen for which signaling takes place. This is a dynamic problem with information uncertainty. The importance of this problem is to show how complex the optimal control strategies may be if only certain information patterns are allowed.

Witsenhausen, H., "A Counterexample in Stochastic Optimal Control," SIAM J. Contr., Vol. 6, 1968.

WITSENHAUSEN'S COUNTER EXAMPLE

- MAJOR CONCEPTS IN GENERALIZED SYSTEM THEORY INCLUDING THE CLASSIFICATION OF INFORMATION STRUCTURES ARE DUE TO WITSENHAUSEN.

- IMPORTANT ILLUSTRATION IS WITSENHAUSEN'S COUNTER EXAMPLE TO THE OPTIMALITY OF LINEAR CONTROLLERS FOR THE LQG PROBLEM WITH NONCLASSICAL INFORMATION PATTERN:

  - TWO CONTROLLERS OPERATING IN SEQUENCE
  - DYNAMICS ARE LINEAR WITH RANDOM I.C.
  - THE COST CRITERION IS THE EXPECTED VALUE OF A QUADRATIC PENALTY ON THE FIRST CONTROL AND THE SECOND CONTROLLER'S DYNAMIC STATE.
  - INFORMATION PATTERN. FIRST CONTROLLER KNOWS DYNAMIC STATE PERFECTLY. SECOND CONTROLLER HAS ONLY A NOISY MEASUREMENT OF DYNAMIC STATE AFTER CONTROLLER ONE HAS ACTED
  - CONTROLLER TWO DOES NOT KNOW CONTROLLER ONE'S CONTROL OR OBSERVATION .
  - RESULT: NONLINEAR CONTROLLERS ARE BETTER THAN BEST LINEAR CONTROLLERS.
  - CONTROLLER ONE TRADES OFF THE COST OF ENHANCING CONTROLLER TWO'S OBSERVATION SIGNAL-TO-NOISE WITH HAVING CONTROLLER TWO REDUCE HIS STATE - SIGNALING.

References

Ho, Y.-C., Kastner, M., and Wong, E., "Teams, Signaling, and Information Theory," IEEE Trans. Auto. Contr., Vol. AC-23, No. 2, April 1978.

Ho, Y.-C., Chu, K.-C., "Team Decision Theory and Information Structures in Optimal Control Problems - Part I," IEEE Trans. Auto.Contr., Vol. AC-17, February 1972.

Sandell, N. and Athans, M., "Solution of Some Nonclassical LQG Stochastic Decision Problems," IEEE Trans. Auto. Contr., Vol. AC-19, April 1974.

Yoshikawa, T., "Dynamic Programming Approach to Decentralized Stochastic Control Problems," IEEE Trans. On Auto. Contr., Vol. AC-20, December 1975.

Kurtaran, B. and Sivan, R., "Linear-Quadratic-Gaussian Control with One-Stop-Delay Sharing Pattern," IEEE Trans. On Automatic Control, Vol. AC-19, pp. 571-574, October 1974.

Kurtaran, B., "Decentralized Stochastic Control with Delayed Sharing Information Pattern," IEEE Trans. Auto. Control AC-21, August 1976.

Variaya, P., and Walrand, J., "On Delayed Sharing Patterns," IEEE Trans. On Auto. Contr., Vol. AC-23, No. 3, June 1978.

Yoshikawa, T., and Kobayashi, H., "Separation of Estimation and Control for Decentralized Stochastic Control Systems," Automatica, Vol. 14, 1978.

SOME EXTENSIONS OF WITSENHAUSEN'S WORK

- HO, KASTNER, AND WONG OBTAINED FURTHER RESULTS ON SIGNALING AND BOUNDS ON THE OPTIMALITY OF SOME SPECIAL PROBLEMS USING INFORMATION - THEORETIC BOUND OF SHANNON.

  - FURTHERMORE, THEY SUGGEST THAT SIGNALING IS A THRESHOLD PHENOMENON. IF

    1. THE COST OF SIGNALING IS TOO HIGH
    2. SIGNAL CHANNEL TOO NOISY
    3. THE UNDERLYING STATE TOO PREDICTABLE

    THEN SIGNALING WILL NOT EXIST.

- DETERMINATION OF INFORMATION STRUCTURES FOR WHICH SEPARABLE CONTROLLERS OCCUR:

  - HO AND CHU INTRODUCED THREE IMPORTANT IDEAS:

    1. PARTIALLY NESTED INFORMATION STRUCTURES.
       EXPANDS THE CLASS OF CLASSICAL INFORMATION PATTERNS.
    2. RADNER'S STATIC TEAM RESULTS.
       RADNER GAVE CONDITIONS UNDER WHICH THE PERSON-BY-PERSON TEAM OPTIMAL SOLUTION IS ALSO THE GLOBAL OPTIMUM.
    3. FOR THE STATIC LQG TEAM PROBLEM THE OPTIMAL DECISION RULES ARE AFFINE.

- Ho and Chu showed that for the partially nested information structure the dynamic LQG problem has an affine decision rule.

- Building on these results, Sandell and Athans, Yoshikawa, and Kurtaran developed sequential solution for the LQG problem with one-step delayed information pattern via dynamic programming, i.e., the control at each node is a function of its present observation and the information at all the nodes one stage behind. The controllers remain affine.

- Kurtaran conjectured that the optimal control for more than one step delay was in the class of separable controls. Counter examples are given by Variaya and Walrand and Yoshikawa.

- Yoshikawa gave a sufficiency condition for separable controls.

- General case of one-step delay information pattern given by Variaya and Walrand. By dynamic programming.

This slide gives the statement of Radner's theorem which is essential for the dynamic decentralized results. This theorem is extremely difficult to verify and although it is a critical result, Radner did not use it in showing that the linear decision rule for the LQG (Linear-Quadratic-Gaussian) Team Problem is globally minimizing, but took a different approach. The object of our work was to relax the local finiteness condition with differentiability conditions that can be verified. In this way, we were able to verify that the positive LEGT (Linear-Exponential-Gaussian-Team) problem is globally optimum. This allows a generalization of the present LQG results.

Radner, R., "Team Decision Problems," Ann. Math. Statist., Vol. 23, No. 3, pp 857-881, 1962.

Krainak, J., Speyer, J., and Marcus, S., "Static Decentralized Team Problems: Sufficient Conditions, Algorithms, and the Exponential Cost Criterion," Proceedings of the 19th IEEE Conference on Decision and Control, December 1980.

Speyer, J., Marcus, S., and Krainak, J., "A Decentralized Team Decision Problem With An Exponential Cost Criterion," IEEE Transactions on Auto. Contr., Vol. AC-25, No. 5, October 1980.

GENERALIZATION OF RADNER'S THEOREM

- THE PROCEDURE HAS BEEN TO REDUCE DYNAMIC PROBLEMS WITH PARTIALLY NESTED INFORMATION PATTERN TO STATIC TEAM PROBLEMS.

- THE HEART OF STATIC TEAM RESULTS IS RADNER'S THEOREM. LET X BE THE "STATE OF THE WORLD," U BE THE CONTROL VARIABLE, $\gamma(\cdot)$ BE THE DECISION RULE $(U = \gamma(X))$, $\psi(X,U)$ BE THE COST OF FUNCTION; AND $J(\gamma) = E[\psi(X,\gamma)]$ BE THE COST CRITERION. IF

    1. $\psi(X,U)$ IS CONVEX AND DIFFERENTIABLE IN $U \; \forall \; X$,
    2. $\text{INF } J(\gamma) > -\infty$
    3. $J(\gamma)$ IS LOCALLY FINITE,
    4. $\gamma = \hat{\gamma}$ SATISFIES THE STATIONARY CONDITIONS,

       THEN $\hat{\gamma}$ IS GLOBALLY MINIMIZING.

    - LOCALLY FINITE: GIVEN THE STATIONARY DECISION RULE, VERIFY THAT ALL PERTURBED FUNCTIONS RESULT IN FINITE COST. THE EXPECTATION OPERATION MAKES VERIFICATION DIFFICULT.

- KRAINAK, SPEYER, AND MARCUS RELAXED RADNER'S CONDITIONS BY REMOVING THE LOCALLY FINITE CONDITION.

    - RADNER GIVES CONDITIONS FOR WHICH PERSON-BY-PERSON OPTIMALITY IMPLIES GLOBAL OPTIMALITY.
    - KRAINAK SHOWS THAT CONVEXITY AND STATIONARITY DIRECTLY IMPLY OPTIMALITY.

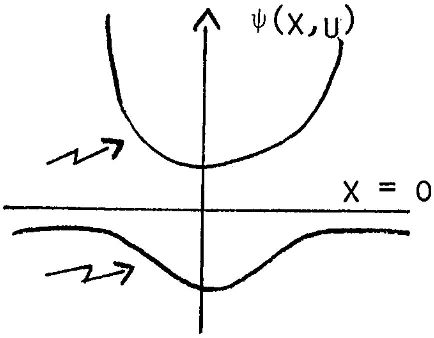
In the following two slides, the LEGT problem performance index is characterized.

SOLUTION TO RADNER'S STATIC TEAM PROBLEM

- TWO FORMULATIONS GIVE LINEAR DECISION RULES:
  - LQG ~ LINEAR - QUADRATIC - GAUSSIAN

    $\underline{\psi}\ (X,U) = X^TQX + U^TNX + U^TRU \quad , \quad R > 0$
  - LEG ~ LINEAR - EXPONENTIAL - GAUSSIAN

    $\psi\ (X,U) = \text{EXP}\ \underline{\psi}(X,U)$



  - IF $U > 0$, POSITIVE EXPONENTIAL
  - IF $U < 0$, NEGATIVE EXPONENTIAL
  - EXPONENTIAL FUNCTIONS ALLOW FOR MULTIPLICATIVE RATHER THAN ADDITIVE COST CRITERION. THIS IS A NATURAL SETTING TO MODEL PROBABILISTIC COST CRITERION USED FOR EXAMPLE IN C^3 PROBLEMS.
- RADNER SOLVES THE LQG PROBLEM AND OBTAINS AN AFFINE DECISION RULE.
  - VERIFICATION OF OPTIMALITY IS DONE THROUGH HILBERT SPACE ARGUMENTS AND NOT BY DIRECT VERIFICATION OF HIS THEOREM.

- AFFINE DECISION RULES SATISFY THE STATIONARY CONDITIONS FOR THE LEG PROBLEM (SPEYER, MARCUS, KRAINAK):

  - THE LEG PROBLEM HAS THE PROPERTY REFERRED TO AS THE "UNCERTAINTY THRESHOLD PRINCIPLE." IF CERTAIN STATISTICAL OR COST PARAMETERS BECOME TOO LARGE, THE COST CRITERION CEASES TO EXIST.

  - FOR THE POSITIVE EXPONENTIAL, ALTHOUGH LOCAL FINITENESS IS NOT VERIFIABLE, AFFINE DECISION RULES ARE SHOWN TO BE MINIMIZING BY THE RELAXED FORM OF RADNER'S THEOREM.

  - SINCE THE NEGATIVE EXPONENTIAL IS NOT CONVEX, FURTHER WORK IS NEEDED TO SHOW THAT THE LINEAR DECISION RULE WHICH SATISFIES THE STATIONARY CONDITIONS IS GLOBALLY OPTIMAL.

  - THE STATIONARY CONDITIONS ARE A K- SET OF COUPLED ALGEBRAIC MATRIX EQUATIONS WITH NUMEROUS POSSIBLE SOLUTIONS.

References:

Speyer, J., "Computation and Transmission Requirements for A Decentralized Linear-Quadratic-Gaussian Control Problem," IEEE Trans. On Auto. Contr., Vol. AC-24, No. 2, April, 1979.

Levy, B. and Castanon, D., "Use of Markovian Models in Information Fusion," MIT/ONR Workshop on C3, Silver Springs, Maryland, June 1980.

Willsky, A., "A Survey of Design Methods for Failure Detection in Dynamic Systems," Automatica, Vol. 12, pp 601-611, 1976.

Shiryayev, A., Optimal Stopping Rules, Springer-Verlag, New York, 1978.

Deckert, T., Desai, M., Deyst, J., and Willsky, A., "F-8 DFBW Sensor Failure Identification Using Analytic Redundancy," IEEE Trans. On Auto. Contr., Vol. AC-22, No. 5, October 1977.

- IN DEVELOPING AIRCRAFT REDUNDANCY SYSTEMS, EACH SENSOR CAN HAVE ITS OWN MICROPROCESSOR WHICH CALCULATES A LOCAL ESTIMATE OF THE STATE.
  - IF THE TOTAL STATE IS NOT OBSERVABLE FROM THE LOCAL OBSERVATIONS, ONLY THE OBSERVABLE PART OF THE STATE IS ESTIMATED ALLOWING REDUCED ORDER ESTIMATORS (LEVY AND CASTANON).
- THESE ESTIMATES SHOULD BE USED FOR DEVELOPING ANALYTIC REDUNDANCY MANAGEMENT SCHEMES (WILLSKY).
  - HYPOTHESIS TESTING METHODS USE THESE ESTIMATES, I.E., THE WALD SEQUENTIAL PROBABILITY RATIO TEST ENHANCED BY AN OPTIMAL STOPPING RULE OF SHIRYAYEV (DECKERT, DESAI, DEYST, WILLSKY).
- IN THE LQG ONE-STEP DELAY INFORMATION SHARING PATTERN, THE DATA THAT IS TRANSMITTED CAN BE COMPRESSED BY DETERMINING THE MINIMUM INFORMATION NEEDED BY EACH CONTROLLER AND STILL ALLOW SEPARABLE CONTROL LAWS.
- ANY TRANSMISSION NOISE PRODUCES NONMECHANIZABLE OPTIMAL NONSEPARABLE CONTROL LAWS. THUS, ONLY SUBOPTIMAL STRUCTURES CAN BE MECHANIZED.
  - INCREASED TRANSMISSION DELAY ALLOWS IMPROVED TRANSMISSION SIGNAL-TO-NOISE RATIO.

LOCAL ESTIMATION, DATA PROCESSING, INFORMATION COMPRESSION, AND FAULT DETECTION

- RATIONALE FOR NETWORK STRUCTURE:
  - SPACIAL DISTRIBUTION OF SENSORS AND ACTUATOR IS IMPOSED FOR THE PURPOSE OF GRACEFUL DEGRADATION UNDER STRUCTURAL PERTURBATIONS.
- FOR THIS PURPOSE, IT MAY BE REQUIRED THAT EACH NODE PROCESS ITS OWN INFORMATION.
- FOR LINEAR - GAUSSIAN MODELS LOCAL KALMAN FILTERS ARE CONSTRUCTED TO PRODUCE CONDITIONAL MEAN ESTIMATES BASED UPON THE LOCAL OBSERVATIONS.
- SUPPOSE THESE LOCAL CONDITIONAL MEAN ESTIMATES ARE TO BE COMBINED TO PRODUCE THE CONDITIONAL MEAN ESTIMATE BASED UPON ALL THE INFORMATION IN THE NETWORK.
- THE GLOBAL ESTIMATE IS A WEIGHTED SUM OF ALL THE LOCAL ESTIMATES PLUS A SUM OF VECTORS THAT MUST BE CALCULATED AT EACH NODE (SPEYER). FORTUNATELY, ONLY A FINITE DIMENSIONAL STATE NEED BE CALCULATED.
- THE DATA PROCESSING NEEDED FOR THIS DISTRIBUTED SYSTEM IS GREATER THAN IF IT WERE CENTRALIZED. HOWEVER, THE DATA IS PROCESSED IN PARALLEL.

RESEARCH DIRECTIONS IN

LARGE SCALE SYSTEMS AND DECENTRALIZED CONTROL

ROBERT R. TENNEY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

I. INTRODUCTION

As the need to manage a variety of large, complex systems is growing, so is the need for a methodology which addresses the issues encountered in design of real time system control. Such issues include dynamical responses to decision variables, uncertainty and errors, desired performance attributes, and complexity.

One candidate discipline for the development of such a methodology is that of control theory, which has, in several cases, successfully merged the first three of the above issues into a framework which can address practical problems. However, the fourth complexity has proven to be a more difficult challenge. This paper will briefly outline the historical development of control theory, the issues it has addressed, and directions which are being explored as the scope of the systems to be managed increases.

II. HISTORY

A. Feedback Control Systems

The roots of control theory lie in the study of small systems such as motors, valves, etc. which have the properties that

a) The system is well understood

b) Certain outputs are measured (which provide information on the internal workings of the system)

c) Inputs can be supplied to it which then affect future outputs

The object of the designer of a control system for such applications was to construct a mechanism (as simple to implement as possible) which derived a set of inputs to be applied as a result of past outputs which had been observed.

A key point which surfaces even at this early stage in the development of the field is that of (a). The system to be controlled was always one which could be analyzed with well-understood physical principles complemented by reasonable engineering insight (e.g. which second-order effects could be neglected). This analysis could lead to a model of the system, often taking the form of ordinary differential equations, which was complete and accurate within some small margin of error.

This leads to a separation of concepts which has persisted to the present. Control theorists assume that, in principle, all of the knowledge pertaining to the relationships of inputs to outputs exists <u>a priori</u> and can be used to guide the selection of future inputs to produce the desired future outputs (i.e. the <u>structure</u> of the task domain is known). There may be residual errors (or noise) in this knowledge due to approximations and simplifications, but then it is the task of the feedback controller to compensate for such errors. Finally, the objective of the control system is well understood - to keep the outputs at or near some nominal value in the face of the errors.

The typical implementation of such a control system consists of

1. Subtracting the nominal values of the outputs (those expected in the absence of errors) from those actually observed
2. Manipulating these error signals with integrators, summers, etc. to produce adjustments to input signals (and deriving this structure and related constants is the designer's task)
3. Adding the adjustments to the nominal inputs (those required to obtain the nominal outputs)

While the above is very implementation-oriented, it begins to illustrate the conceptual issues involved in any real time decision system. One can view the control system as making a series of decisions as to which inputs to supply at each time, based upon information obtained through observations of the error signals. The key difference between this and knowledge-based approaches to decisionmaking is the fact that the designer's understanding of the system structure has been distilled to such an extent that very little is left: namely, that which directly impacts the computation of the next input. The understanding of the system is incorporated in the off-line development of the controller, which in turn only processes the new information which appears in the on-line setting. This leads to the conclusion that basic feedback control techniques cannot be applicable to problems where the system structure changes (e.g. as a result of component failures, additions to it, wear, etc.)

These initial developments were thus limited as automatic decision systems, yet began to deal with the issues of feedback (which can drastically alter qualitative system behavior).

Progress proceeded in three different directions from here:

1. Classical frequency domain control theory, using transform techniques and thus limited to systems modelled by linear differential equations (and, until recently, to those with single inputs and single outputs)
2. Modelling of systems, in which quite large systems could be modelled (primarily for simulation purposes). This includes the development of statistical techniques for deriving the system structure from observed data
3. Modern control theory, in which the mathematical concepts of state, probability, and optimization are combined into a framework which yields control decisions which are, in a special sense, the "best" possible

Of these, the first two are fascinating but only indirectly related to the problem of automated decisionmaking to control large or complex systems, but the latter is of direct interest.

B. Modern Control Theory

By precisely defining and using the notion of the state of a system (all knowledge of the past relevant to determining the effect of future inputs), modelling the system as a set of transformations (one of which is selected by the inputs) on that state, and seeking to find controls which optimize some performance index, quite a bit can be deduced about the structure of a suitable controller for the system. That the formulation of a control problem in these terms is about identical to that of early problem-solving methods (e.g. GPS by Newell, Shaw & Simon) is probably not coincidental. Both were attacking the same question: how to manipulate a world to achieve desired ends.

The difference in the two approaches comes in the kinds of additional assumptions on system structure which have been made to allow their practical application (both solve any decision problem in principle). Modern control theory is best applied in contexts of systems with "simple" structure, in which either the number of states is small enough to allow each to be considered explicitly (e.g. queueing systems) or the structure is such as to allow a multitude of states to be considered implicitly through symbolic manipulations (e.g. least-squares best fit in target tracking problems).

In decisionmaking terms, the fundamental impact of this way to formulate the problem comes as a result of a separation principle: it is sufficient (and usually necessary) for the controller to know the current state of the system (or a probability distribution on it) in order to derive the optimal control. This leads to an implementation which falls into two pieces: (1) a state estimator which both keeps track of the evolution of the system state as a result of inputs applied, and uses information gained through observation of outputs to update its estimate of that state, and (2) a controller which chooses an input to apply given the state information supplied by the estimator. Since the state estimator must operated using real time observations, its operation cannot be predetermined and hence it must contain a model of the system on-line in order to perform its function. (The controller, however, can be determined beforehand and thus implemented essentially as a table lookup or deterministic function evaluation).

The strengths of the modern control theoretic approach are chiefly:

1. It deals with several facets of the decision problem (multiple inputs and outputs, uncertainty, performance goals) in one framework
2. It allows sophisticated analytical techniques to be brought to bear which exploit the structure of a problem
3. It supports the precise definition of concepts involved, their interrelationships, and thus techniques for validation of assertions about the decisions being made

However, the moderncontrol methodology faces severe limitations due to some serious drawbacks, particularly when one contemplates applying it to large, complex, or distributed systems.

1. The tools for dealing with the above issues are not always suitable for an application (e.g. probabilistic models are unavailable, or a meaningful performance measure is not apparent).
2. The analytical tools are often applicable to very narrow classes of systems and much insight is required to formulate the control problem in terms which fall into one of these classes.

3. Practical application of the analysis to complex systems is severely limited by the computational resources required, even if the control problem is solvable in principle.

Thus modern control theory can be viewed as a library of analytical techniques which are suitable for special classes of problems, and the control engineer is faced with the problem of selecting those which best suit a problem at hand. Recent research has gone into extending this catalog in at least four areas:

1. Relating modern control ideas to those developed in the classical setting
2. Bringing in notions of estimating the system structure, as well as its state, on-line
3. Extending the formulation to include multiple decision makers with different performance criteria but identical information about the system (games)
4. Dealing with issues of complexity directly, perhaps using several controllers with differing information about the system

The first three are exciting areas of work as they combine different approaches to a problem, yet they do not address the issues of complexity directly (the identical information assumption in (3) precludes the "cooperating experts" viewpoint which is included in (4)). The fourth, however, deals directly with them.

C. Large Scale System Theory

Large scale system theory is inherently an ill-defined area, as the definition of "large" changes with each advance in computing technology. A working definition has been that a system is large if none of the standard control techniques can be applied due to practical constraints. Examples include control of the U.S. electric power system, transportation networks, manufacturing systems, large space structures, and command and control systems.

One approach to controlling a large system is to decompose it into weakly interacting, small subsystems. Quite a bit of current research is moving along these lines, as new ways to exploit nascent system structure are found. Two of these include spatial separation, in which several parts of a system each impact others in a way which is minor compared with its internal working (e.g. geographically separated traffic systems), and temporal separation, in which the rates of change of two parts of a system are so diverse that one looks like noise to the other, and the latter like a constant to the former. Again, though, these approaches are constrained to apply only to problems which can be posed so as to display the requisite structure.

A second approach is to postulate the existence of several controllers, either cooperating as a team or working individually, but which have different information being received about the system. Each may observe different outputs, be responsible for different inputs, and be responsible for different aspects of the system's behavior. This decentralization of decision-making power is motivated by either:

1. Constraints on the implementation (communications from all sensors to one site would be prohibitively expensive)
2. An approach to handling complexity through the divide and conquer concept

Research in decentralized control which has followed the lines suggested by the first of these has been fairly successful. For certain feasible communications policies (i.e. for a fixed, known control system structure) problems can be formulated which reduce down to parameter optimization problems. This somewhat begs the question of what communications structure should be used, and focuses on how to use that which one may have.

Unfortunately, decentralization in modern control theoretic framework has not proven to be successful in handling complexity - in fact, just the opposite is true. Even superficially simple problems become quite complex when approached this way, but it is becoming clear why.

1. Optimization strongly encourages centralization - it is clearly better (if one assumes unbounded processing power) to have one decision maker receive all information and make decisions based on it, than to have the information in separate places. Thus care has to be taken to formulate a decentralized decision problem to prevent escape hatches through which the equivalent of centralization can occur.
2. As a corollary to this, it may be reasonable for one decision maker to use the system being controlled as a communications channel to transmit extremely crucial information to another decision maker. Thus its decisions are not only directed towards achieving desired performance of the system, but also to communicating and thus affecting performance indirectly. Such <u>signalling</u> strategies are not necessarily undesirable, but they greatly complicate the analysis required to understand decentralization.
3. There is a problem in that each decision maker needs not only to consider the impact of its decisions on the system directly, but also to consider its impact in the context of inputs being supplied by other decision makers. This naturally requires some

ability to predict what those other inputs are likely to be, and thus some knowledge of the strategies used by the other decision makers. This <u>second-guessing</u> argument is particularly bothersome when it recurses: each of the other controllers must model the one decision maker, which includes the latter's knowledge of the former's knowledge of the latter .....

The above problems encountered in the straightforward extension of modern control theoretic ideas to decentralization may not be insurmountable (and evidence is accumulating that the second-guessing problem can be resolved), but they do make it worthwhile to consider new approaches to complex control problems which draw on the capabilities and advantages of established control theoretic techniques, yet allow the consideration of complexity in a more natural framework. Such new approaches demand imagination and innovation, and some tentative forays into defining some will be mentioned next.

However, it is worthwhile to summarize the advantages of modern control approaches to real time control of systems which have been presented above.

1. The most successful applications of control theory have been in contexts where the system displays simple structure which can be well modelled.
2. For those situations where a single performance criterion can be defined, these techniques always generate decisions which are consistent in the system-wide sense, and which use all available information from all services in the most effective way possible.
3. For those situations where meaningful probabilistic descriptions of uncertainty can be constructed, the decisions reflect that

uncertainty and take into account the fact that more information will be available in the future.

4. Control theory is well suited for dealing with dynamical systems, where the effects of a decision propagate through time.
5. Through a combination of engineering insight and mathematical analysis, automated decision systems can be built which quite successfully operate in real time.

III. RESEARCH DIRECTIONS

While there is no clear path to gain understanding of the issues of complexity and decentralization, there is considerable feeling that such understanding can be obtained. It is important that any approach to automated decisionmaking in a complex environment deal squarely with the relevant issues of dynamics, uncertainty, system-wide performance goals, and complexity, preferably in a unified and well defined framework. Three approaches which are currently being actively pursued are:

1. Continuation of the development of special techniques for problems with special structure. By working through a series of special cases, each progressively richer in detail than the last, the available understanding of interrelations of decisions, particularly in the decentralized case, can be increased.
2. Hybrid approaches:there is a great possibility that a judicious combination of control theoretic techniques with those of high level problem solving as developed by the artificial intelligence community can provide a great payoff. The specialized control theoretic techniques might be viewed as a lower level of decision making backed up by, and supporting, higher level knowledge processing. For example, the high level processes might generate

plausible courses of action which can then be augmented by the low level control derivations. Similarly, the higher level might generate hypotheses as to what is occurring within the system, while lower level statistical techniques evaluate them in light of specific sensor information (in fact, such an architecture has already been developed in a few applications such as EKG analysis and aircraft tracking systems). Essentially, the high level, knowledge-based processing can deal with the combinatorial explosion generated by complex decision domains, while the lower level techniques can fine tune the result. The complementarity of control theoretic and AI techniques, as well as the similarity of their essential concepts, bodes well for research in this area.

3. Mathematical modelling of knowledge: control theorists have developed ways to define concepts of structure, performance, and information, but have yet to deal with the concept of the knowledge available to a decisionmaker. However, both in the hybrid approach to complexity and in the case of decentralization, this seems to be a major issue. A goal of such research might be to formulate models of knowledge which are amenable to the same sorts of manipulation as information, performance, etc. Some very preliminary ideas are emerging along this line, but it will be some time before they can be put into perspective.

In conclusion, then, it seems that a major thrust of future research in large scale systems and decentralized control will and should be along lines devoted to blending knowledge-based processing with more analytical control techniques. Such work should expand the scope of problems which can be solved with an automated decision system, and be useful in a wide variety of practical settings.

MAJOR POINTS:

1. CONTROL THEORY PROVIDES A WELL-ESTABLISHED FRAMEWORK FOR DEALING WITH AUTOMATIC DECISION PROBLEMS

2. CONTROL THEORY PROVIDES A SET OF TECHNIQUES FOR AUTOMATIC DECISION MAKING WHICH EXPLOIT SPECIAL STRUCTURE

3. CONTROL THEORY DOES NOT DEAL WELL WITH COMPLEXITY

4. POTENTIAL EXISTS FOR COMBINING CONTROL THEORETIC AND KNOWLEDGE-BASED CONCEPTS INTO A UNIFIED APPROACH

OUTLINE

ROADMAP OF CONTROL THEORY

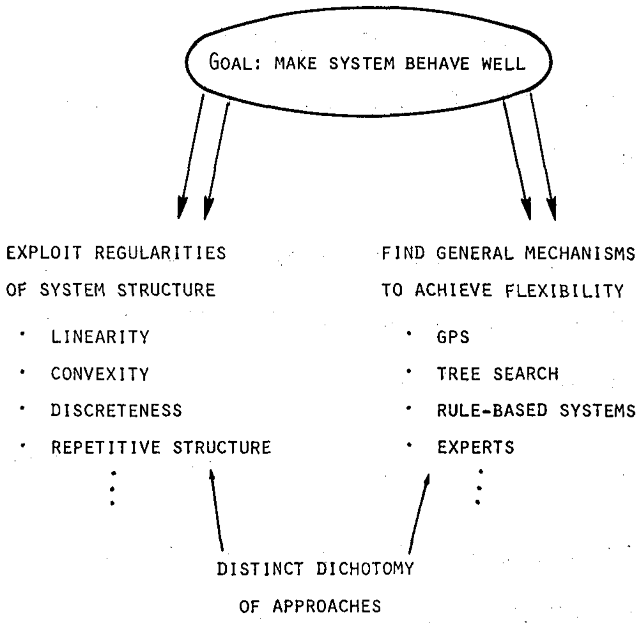
- CLASSICAL
- MODERN
- LARGE SCALE

CONTROL OF LARGE SCALE SYSTEMS

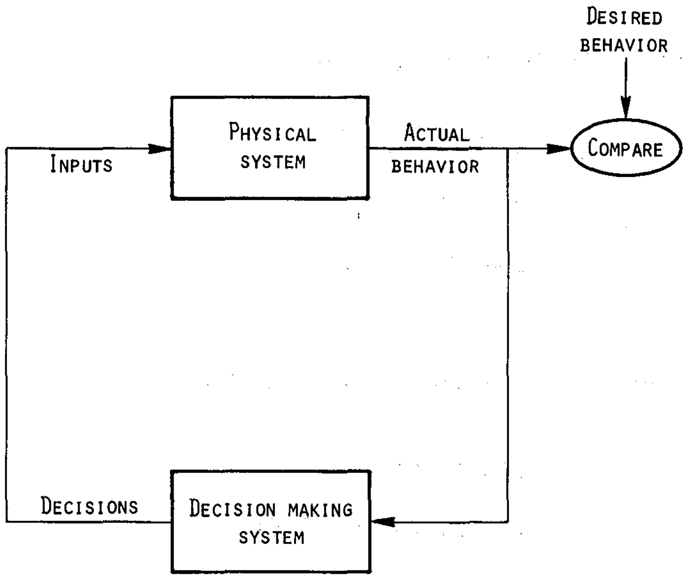
- DECOMPOSITION
- DECENTRALIZATION
- HYBRID APPROACHES

RESEARCH DIRECTIONS

TWO APPROACHES TO AUTOMATIC DECISION MAKING:

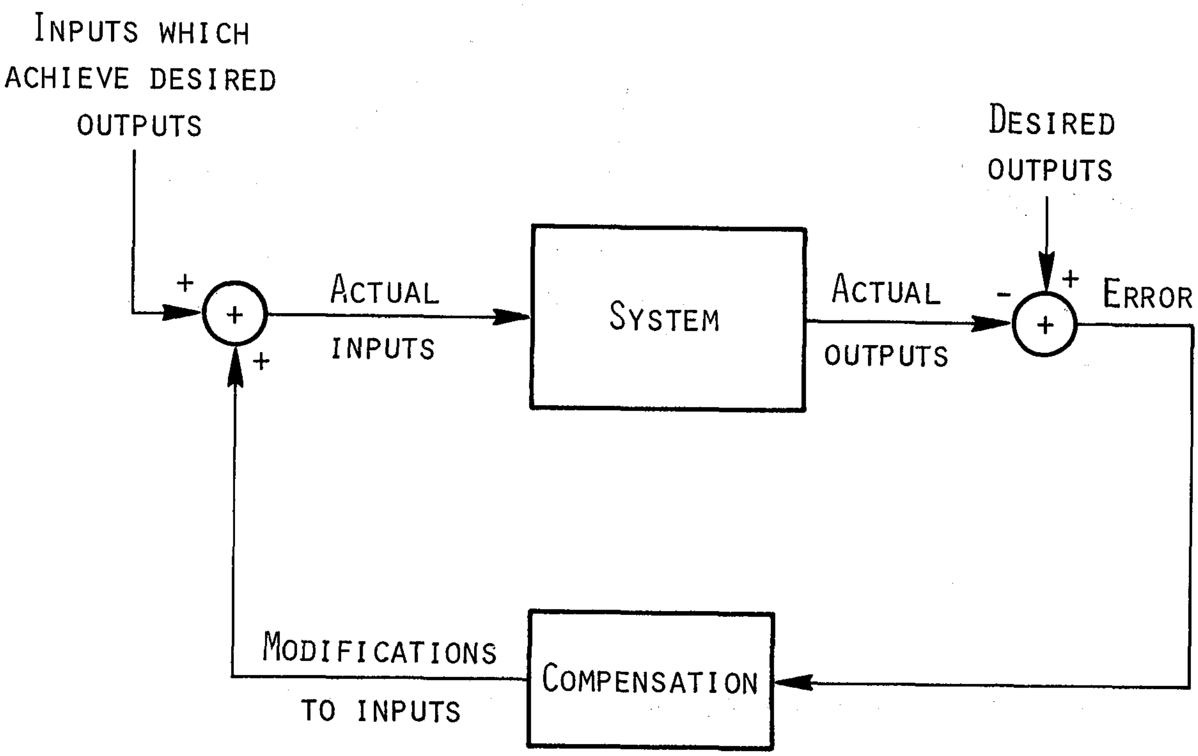


GENERAL VIEW OF CONTROL THEORY:



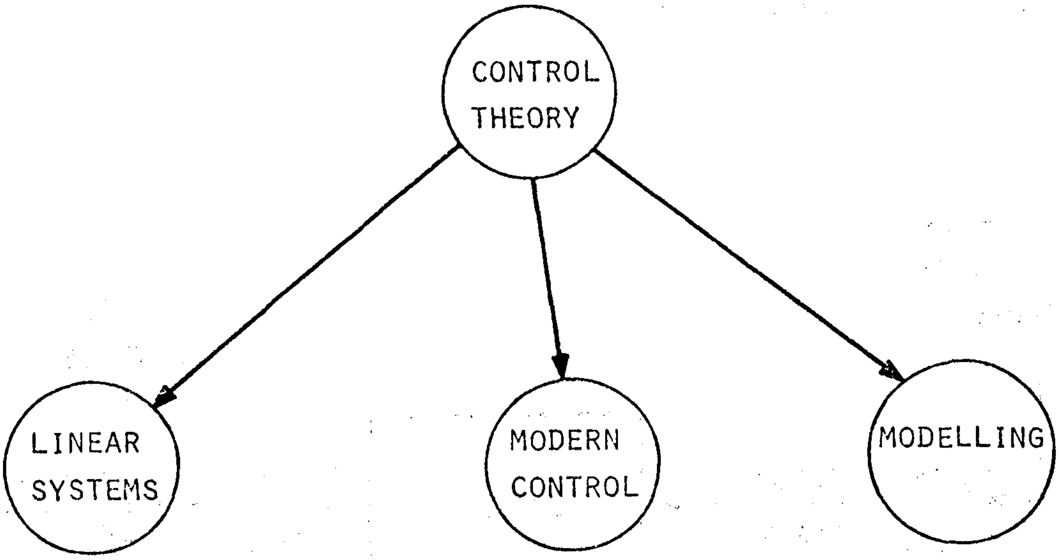
GOAL: DESIGN DECISION MAKING SYSTEM TO CAUSE PHYSICAL SYSTEM TO DISPLAY DESIRED BEHAVIOR

BACKGROUND OF CONTROL THEORY:



GOAL: DESIGN COMPENSATION TO BE

1. EFFECTIVE IN MAINTAINING DESIRED OUTPUT
2. SIMPLE TO IMPLEMENT



LINEAR SYSTEMS THEORY

- DEVELOPMENT OF SPECIAL UNDERSTANDING OF, AND ANALYSIS AND DESIGN TECHNIQUES FOR, "SMALL" LINEAR SYSTEMS

$$\dot{\vec{x}} = \underline{A}\vec{x} + \underline{B}\vec{u}$$

$$\vec{y} = \underline{C}\vec{x}$$

- ORIENTED AROUND FREQUENCY DOMAIN TECHNIQUES
- IMPORTANT AS LINEARITY IS A COMMON PROPERTY OF (APPROXIMATIONS TO) PHYSICAL SYSTEMS

MODELLING

- TECHNIQUES TO CONSTRUCT MODEL OF A SYSTEM GIVEN INPUT/OUTPUT DATA

- STATISTICS AND OPTIMIZATION ARE MAJOR COMPONENTS

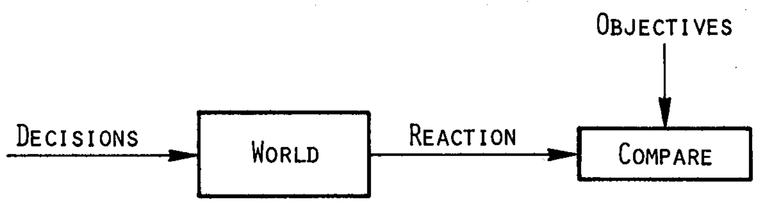
- RELY ON ENGINEERING INSIGHT AS TO TYPE OF MODEL

- CAN BE APPLIED TO LARGE SYSTEMS, PARTICULARLY IN SIMULATION

ELEMENTS OF MODERN CONTROL THEORY

1. OPTIMIZATION - SERVES TO CONVEY OBJECTIVES OF DECISION SYSTEM

$\Longrightarrow$ DO AS WELL AS POSSIBLE



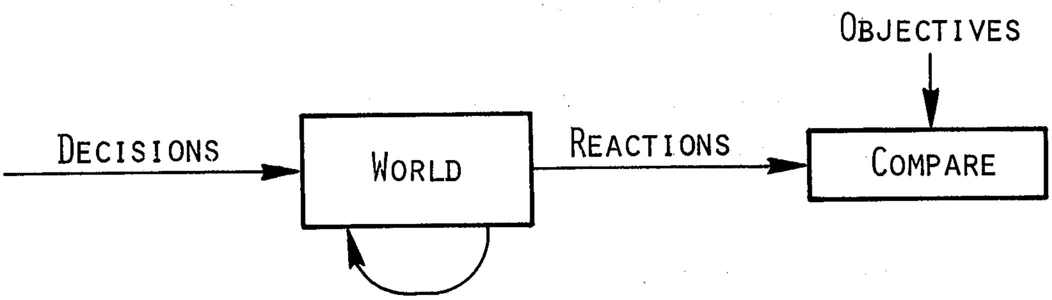
ASSUME REACTIONS CAN BE ORDERED WITH RESPECT TO ONE ANOTHER (QUANTIFIABLY)

EXAMPLES: MINIMIZE FUEL USE
TIME
RMS ERROR
PROBABILITY OF ERROR

NOTE: PERFORMANCE MEASURES OFTEN INCOMPLETE

2. DYNAMICS - DECISIONS AFFECT ONE ANOTHER, AND THEIR EFFECTS PROPAGATE THROUGH TIME

$\Longrightarrow$ PLAN AHEAD



ASSUME THERE IS SOME MODEL OF THE WORLD (STATE SPACE FORM) WHICH RELATES:

STATE AND INPUTS AT TIME t

TO

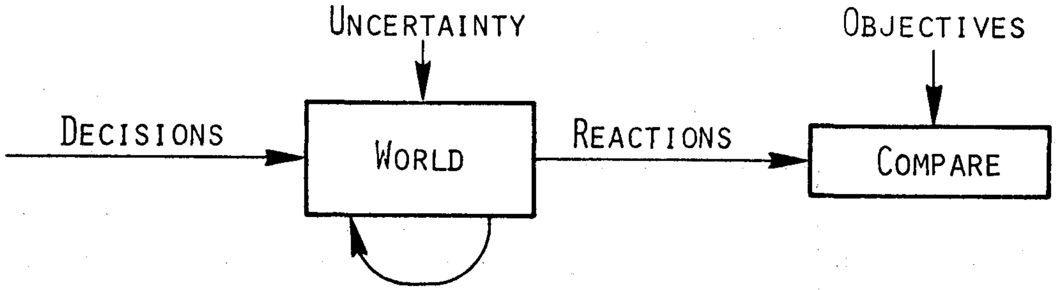
STATE AND OUTPUTS AT TIME $t + \delta t$

EXAMPLE: $\dot{\vec{x}} = \vec{f}(\vec{x}, \vec{u}, t)$

- ROCKET TRAJECTORY
- VOLTAGES AND CURRENTS

NOTE: MODEL USUALLY ASSUMED TO EXIST BEFORE CONTROL SYSTEM IS DESIGNED

3. UNCERTAINTY - DUE TO UNMODELLED PHENOMENA
ERRORS, NOISE
$\Longrightarrow$ HEDGE



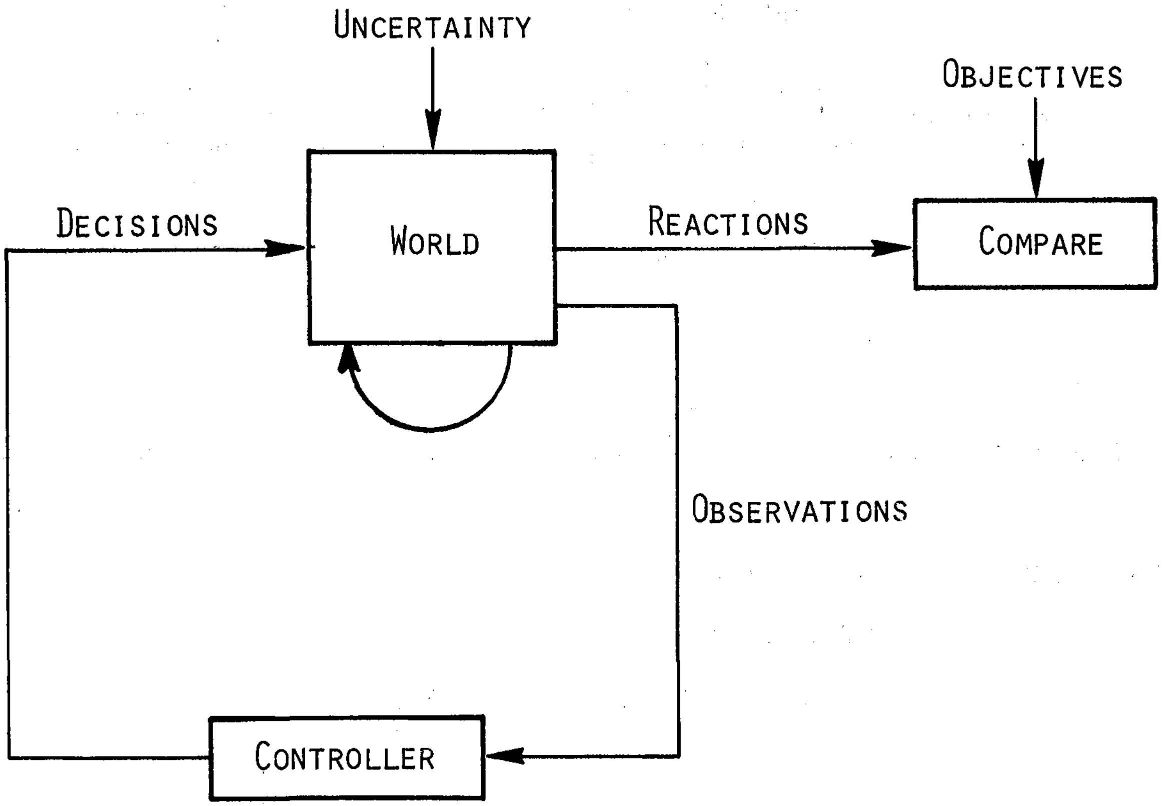
UNCERTAINTY CAN BE DESCRIBED PROBABILISTICALLY OR AS AMBIGUITIES

EXAMPLES:
- TRACKING MOVING TARGETS
- COMMUNICATION IN NOISE
- CONTROL OF SYSTEMS WHICH FAIL

NOTE: NEED FOR DESCRIPTION OF UNCERTAINTY GREATLY INCREASES MODELLING EFFORT

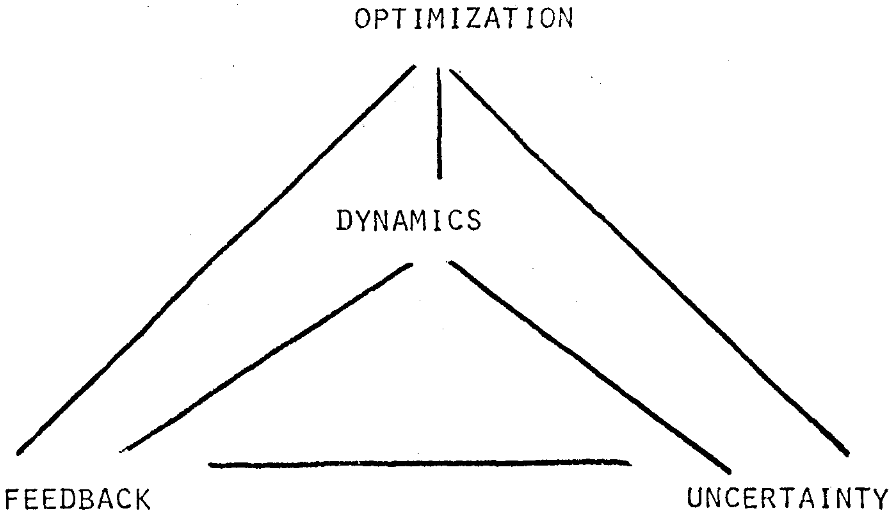
4. FEEDBACK - OBSERVATIONS ARE AVAILABLE WHICH PROVIDE INFORMATION WHICH HELPS RESOLVE UNCERTAINTY

$\Longrightarrow$ REFINE PLANS AS TIME PASSES



NOTE: FEEDBACK CAN DRAMATICALLY ALTER SYSTEM BEHAVIOR

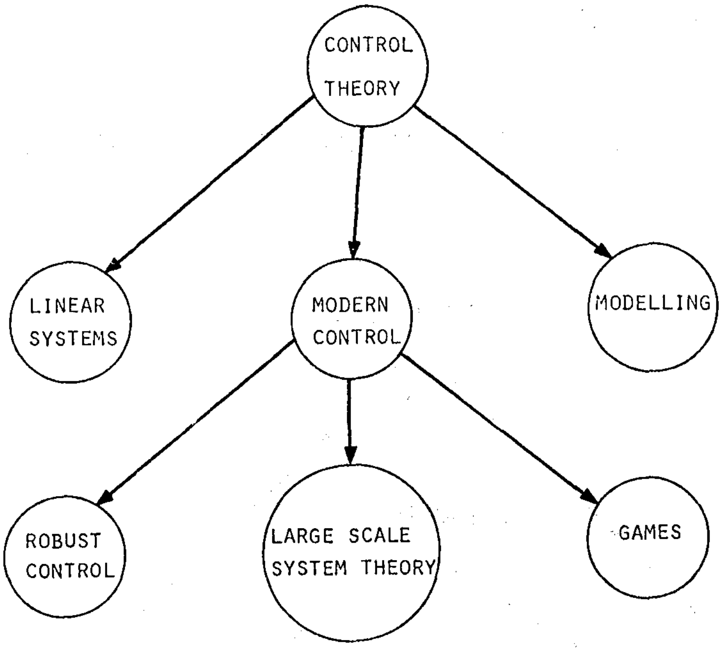
STOCHASTIC OPTIMAL DECISION MAKING



IF ANY INTERESTING DECISION PROBLEM DISPLAYS THESE FOUR CHARACTERISTICS,

THEN MODERN CONTROL OFFERS

1. A WELL DEFINED FRAMEWORK FOR ADDRESSING THEM
2. A SET OF TECHNIQUES WHICH YIELD INSIGHT INTO THE <u>STRUCTURE</u> OF A GOOD DECISION SYSTEM
3. SOLUTIONS WHICH CAN BE IMPLEMENTED IN REAL TIME FOR SYSTEMS WITH SPECIAL STRUCTURE

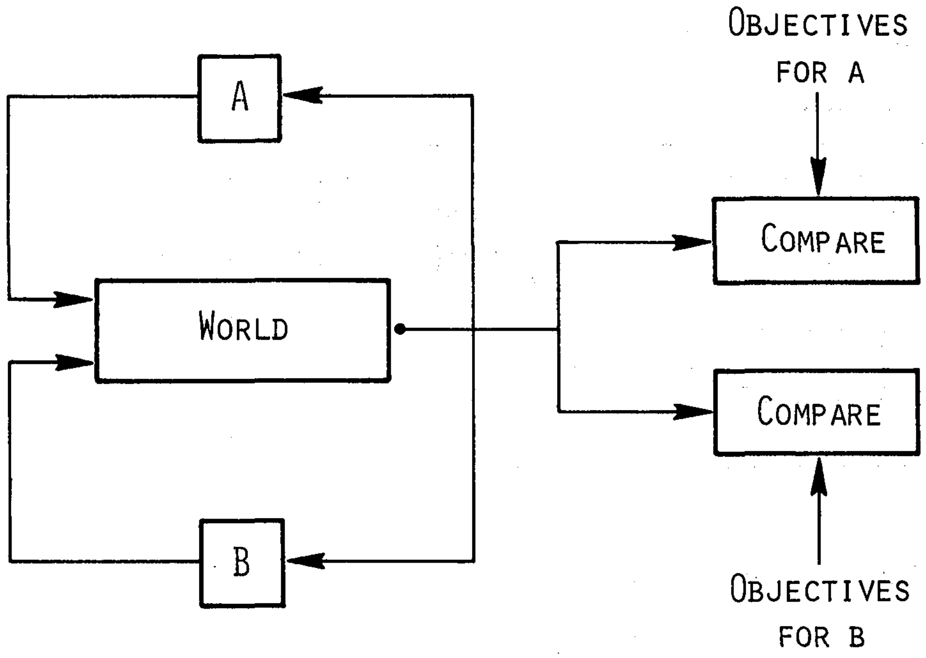


ROBUST CONTROL

- DESIGN CONTROL SYSTEM WHICH IS INSENSITIVE TO ERRORS IN MODEL DUE TO
  - INACCURATE DATA
  - APPROXIMATIONS
  - SLOW CHANGE (WEAR)

- APPROACHES:
  1. DESIGN CONTROLLER FOR CLASS OF SYSTEMS "CONTAINING" THE ACTUAL ONE
  2. ADAPT THE CONTROLLER TO CHANGES IN THE SYSTEM

DYNAMIC, STOCHASTIC GAMES



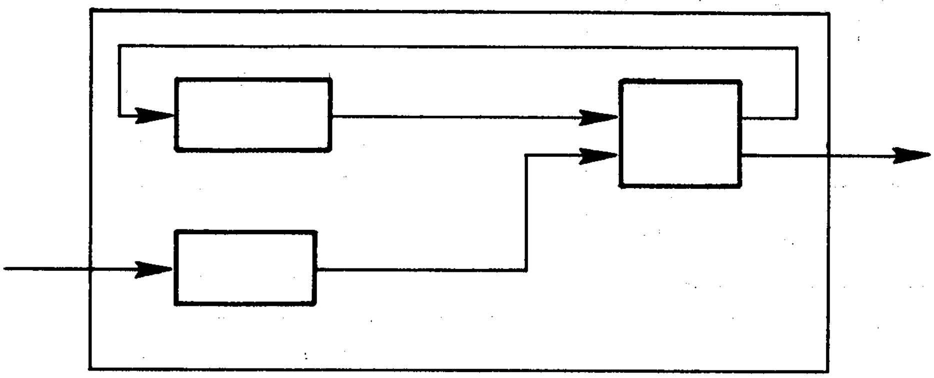
ASSUME MULTIPLE DECISIONMAKERS WITH

1. IDENTICAL INFORMATION (MODEL AND OBSERVATIONS)
2. DIFFERENT OBJECTIVES

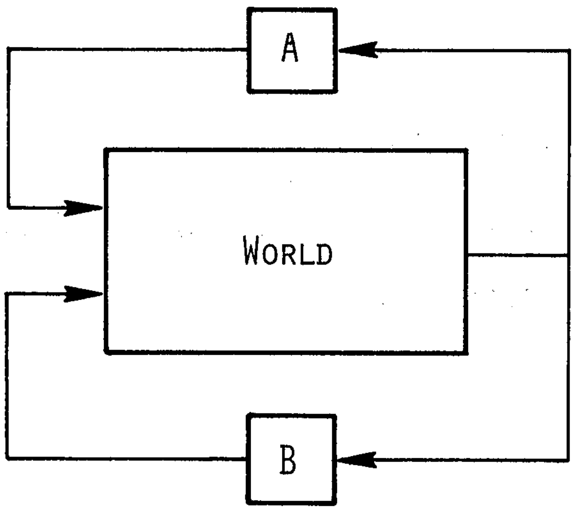
LARGE SCALE SYSTEM THEORY

ATTEMPTS TO DEAL WITH SYSTEMS FOR WHICH SOLUTION BY NORMAL MODERN CONTROL TECHNIQUES IS IMPRACTICAL (DUE TO COMPLEXITY)

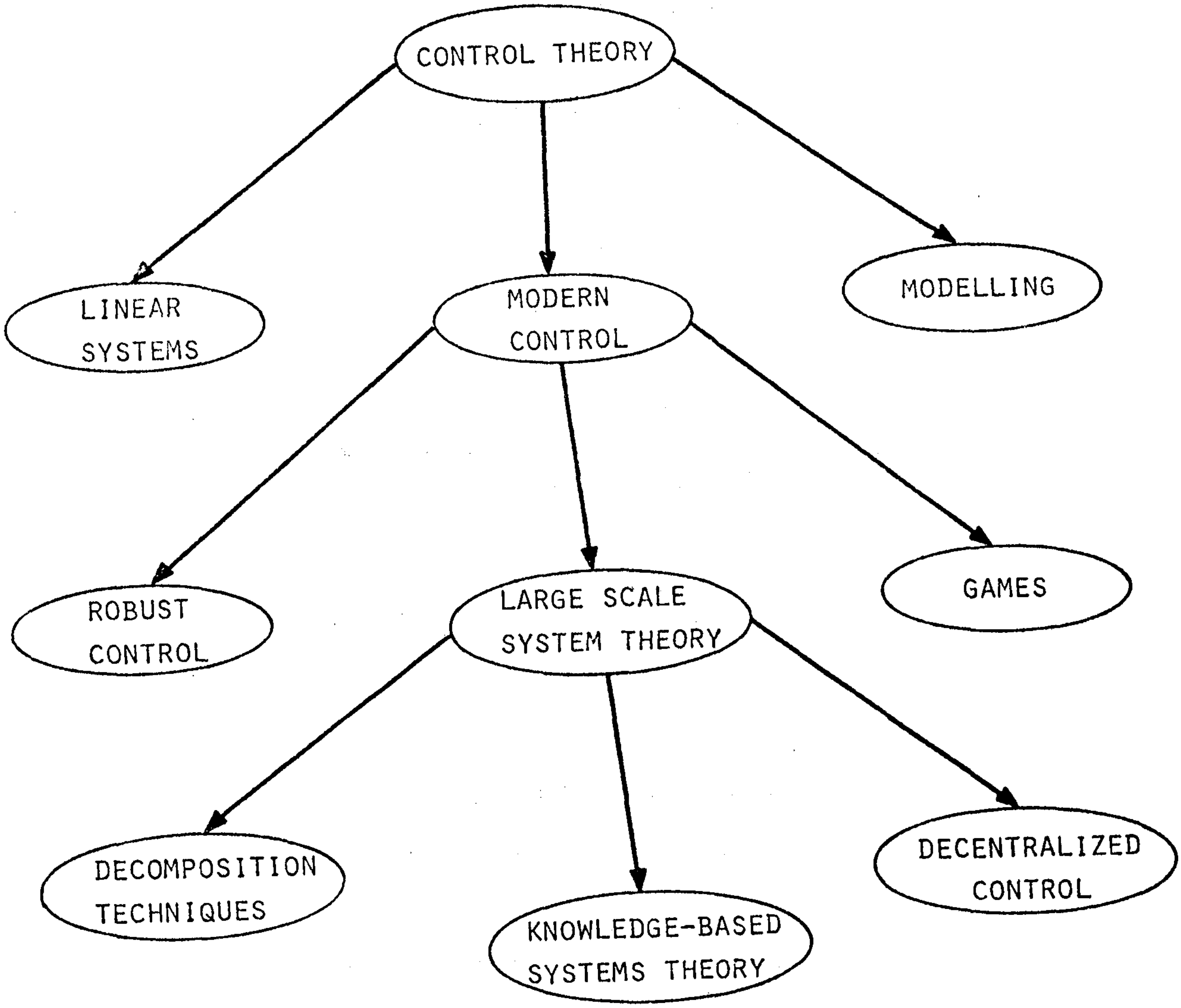
1. DECOMPOSITION



2. DECENTRALIZATION



3. HYBRID APPROACHES

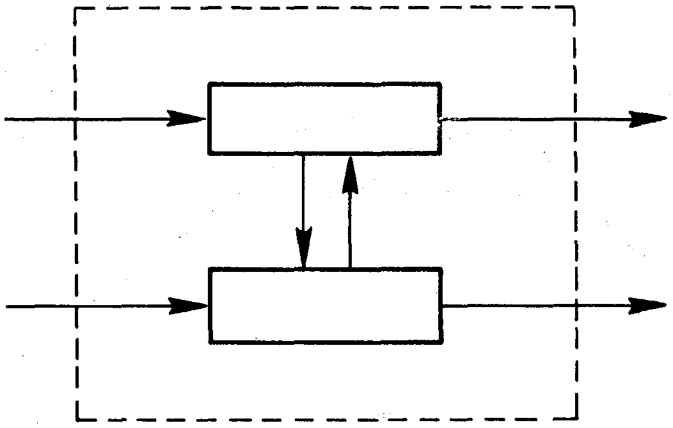

CONTROL THEORY
LINEAR SYSTEMS
MODERN CONTROL
MODELLING
ROBUST CONTROL
LARGE SCALE SYSTEM THEORY
GAMES
DECOMPOSITION TECHNIQUES
KNOWLEDGE-BASED SYSTEMS THEORY
DECENTRALIZED CONTROL

LARGE SCALE SYSTEM THEORY

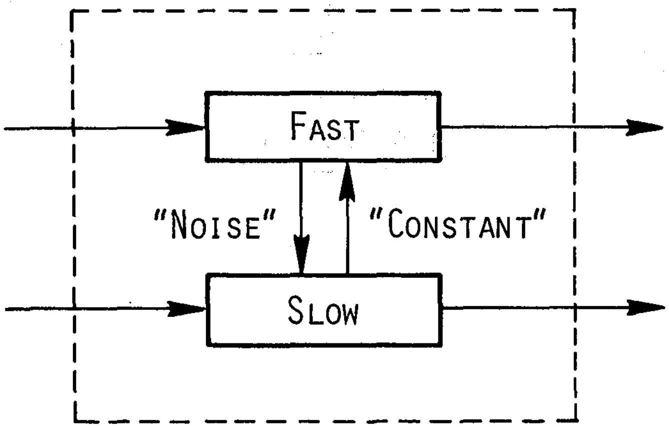
1. DECOMPOSITION = "DIVIDE AND CONQUER"

EXPLOIT STRUCTURES SUCH AS

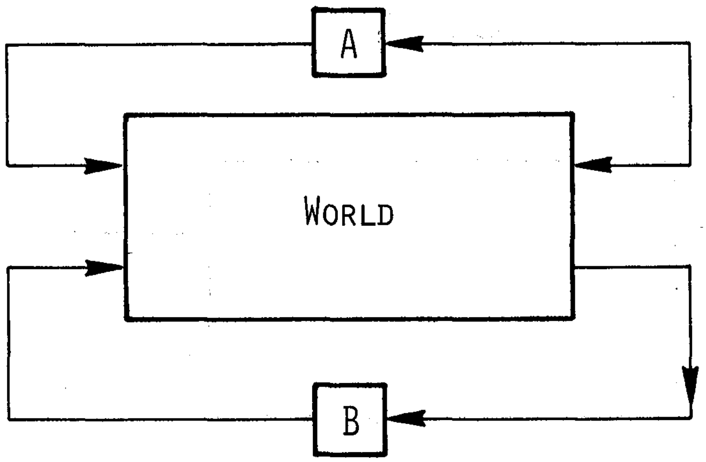
A. WEAK SPATIAL COUPLING



B. WEAK TEMPORAL COUPLING



2. DECENTRALIZATION - USE SEVERAL DECISION MAKERS TO SHARE EFFORT OF CONTROL EACH "SIMPLER" THAN EQUIVALENT CENTRALIZED CONTROLLER



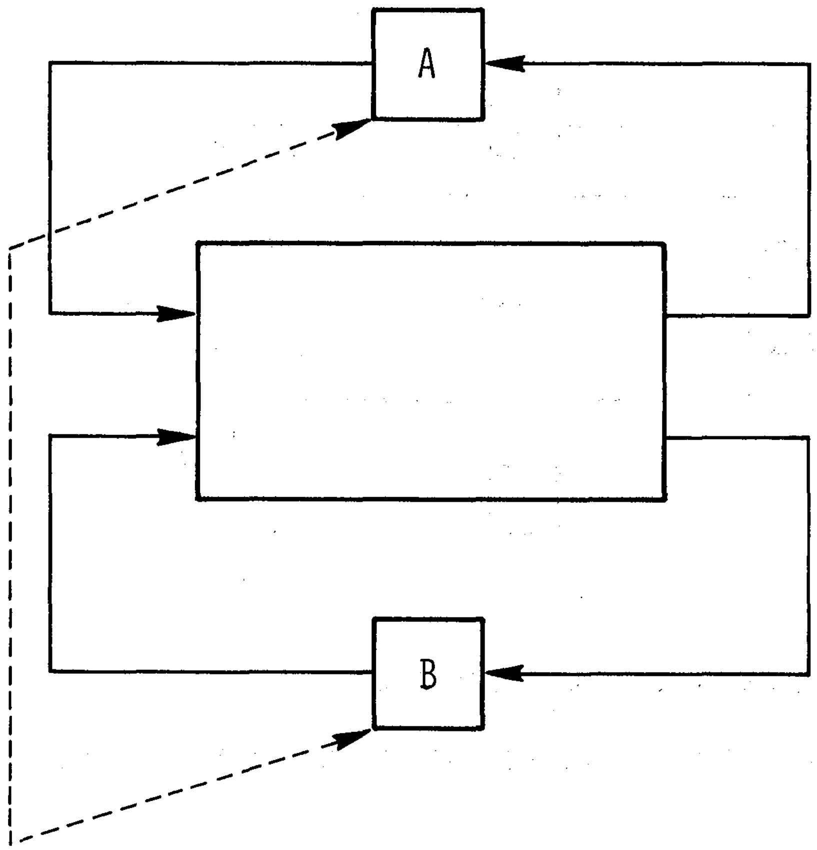
ASSUME

1. ALL SHARE COMMON OBJECTIVE
2. EACH RECEIVES DIFFERENT INFORMATION (SPECIALIZATION)

PARADOX: THIS MAKES PROBLEM MUCH MORE COMPLEX!

1. OPTIMIZATION STRONGLY ENCOURAGES CENTRALIZATION

ANALYSIS SEEKS WAYS TO ACHIEVE COMMUNICATION



B MUST NOT SEND A ALL OBSERVATIONS OR A WILL ACT AS CENTRAL DECISION MAKER

- FORMULATE WITH CARE
- LIMIT COMMUNICATIONS

DECENTRALIZED CONTROL

2. SOLUTIONS MAY BE RANDOMIZED

EXAMPLE: INTERRUPTED PHONE CALL

		A: X	A: Y
B	X	BAD	GOOD
	Y	GOOD	BAD

WITH NO PRIOR COMMUNICATION, EACH CHOOSES X OR Y WITH PROBABILITY 1/2

TWO STEP DECISION PROCESS:

1. SET UP PROTOCOLS (PRIOR)
2. USE THEM (REAL TIME)

DECENTRALIZED CONTROL

3. SECOND GUESSING

A'S MODEL OF EXTERNAL WORLD INCLUDES B.

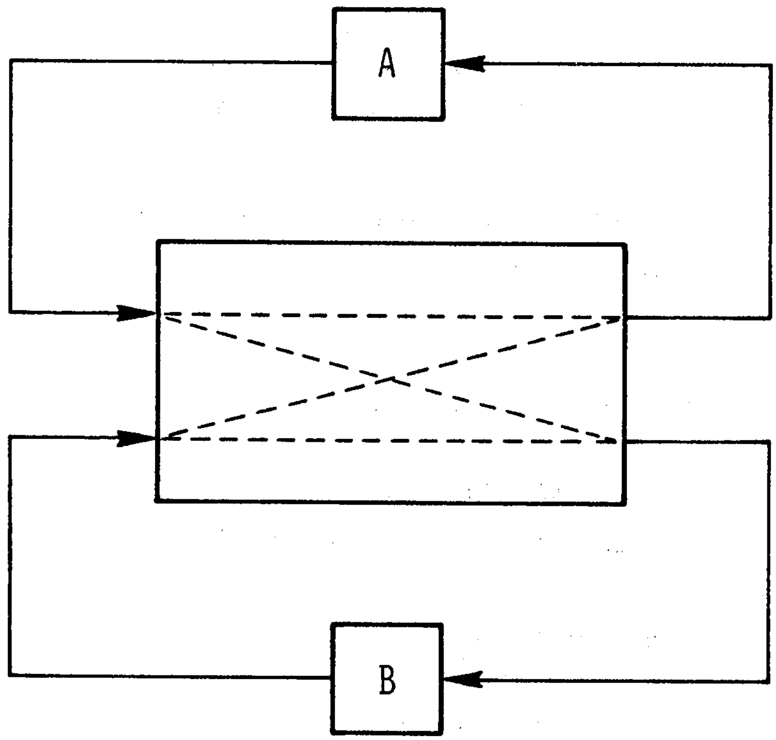
B'S MODEL OF EXTERNAL WORLD INCLUDES A, AND A'S MODEL OF B.

A'S MODEL OF EXTERNAL WORLD INCLUDES B, AND B'S MODEL OF A, AND B'S MODEL OF A'S MODEL OF B.

•
•
•
•

CONVERGENCE ? MAYBE!

4. SIGNALLING



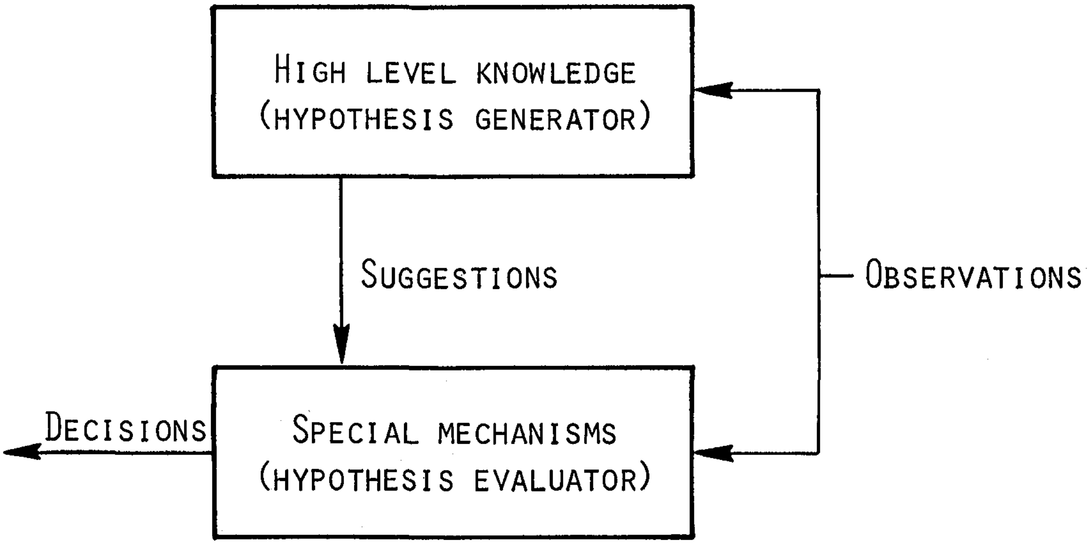
A <u>CAN</u> USE SYSTEM DYNAMICS AS COMMUNICATION CHANNEL TO B

∴ A'S DECISIONS INCLUDE OPTIONS WHICH COMBINE CONTROL WITH COMMUNICATION

SIGNALLING DEGRADES PERFORMANCE DUE TO DIRECT IMPACT OF A, BUT MAY IMPROVE IT INDIRECTLY THROUGH B

3. HYBRID APPROACHES - COMBINE CONTROL THEORETIC AND ARTIFICIAL INTELLIGENCE CONCEPTS IN MUTUALLY BENEFICIAL WAY

A. BILEVEL SYSTEM



EXAMPLE: EKG ANALYSIS
MULTIOBJECT TRACKING

HYBRID APPROACHES

B. MANIPULATION AND MODELLING OF KNOWLEDGE

1. UNDERSTAND ISSUES SUCH AS

- PERSPECTIVE
- EXPLANATION
- QUERIES
- LIMITED COMMUNICATION/MEMORY/PROCESSING WHEN SEVERAL DECISION MAKERS EXIST

2. DECOMPOSE KNOWLEDGE IN REASONABLE WAY (ALGEBRAIC ?)

EXAMPLES: MULTIOBJECT TRACKING
POWER SYSTEM SECURITY
COMMAND AND CONTROL SYSTEMS

RESEARCH DIRECTIONS

A. ANALYTIC

DECOMPOSITION: DEVELOP NEW TECHNIQUES FOR DECOMPOSING OR SIMPLIFYING PROBLEMS

DECENTRALIZATION: DEVELOP APPROACHES TO SPECIAL CLASSES OF PROBLEMS TO HANDLE

- SECOND GUESSING
- SIGNALLING
- PROTOCOLS

MODELLING: DEVELOP NEW CLASSES OF MODELS WHICH ARE SUSCEPTIBLE TO ANALYSIS

IN SHORT: EXPAND SET OF SPECIAL TECHNIQUES WHICH CAN BE BROUGHT TO BEAR ON PROBLEMS

RESEARCH DIRECTIONS

B. STRUCTURAL

EXPLOIT COMPLEMENTARITY OF AI AND CONTROL THEORY AS MODELLING AND DECISION MAKING TECHNIQUES

DEVELOP WAYS TO EXPRESS, MODEL, AND ANALYZE THE CONCEPT OF KNOWLEDGE IN A FRAMEWORK OF MULTIPLE, COOPERATING DECISION MAKERS

COMBINE THESE INTO METHODOLOGY WHICH DEALS WITH DYNAMICS, UNCERTAINTY, OBJECTIVES, FEEDBACK, COMMUNICATION, AND DISTRIBUTION OF CONTROL, INFORMATION AND KNOWLEDGE IN A UNIFIED FRAMEWORK

SYSTEMS MODELING PAST, PRESENT, AND FUTURE AS VIEWED FROM A NETWORK MODELING PERSPECTIVE

GARY E. WHITEHOUSE
UNIVERSITY OF CENTRAL FLORIDA

In this talk, the general area of stochastic networks will be discussed. In particular, I plan on discussing GERT (Graphical Evaluation and Review Technique). This is a procedure which combines the disciplines of flowgraph theory, moment generating functions, and PERT to obtain the solution to stochastic problems. It has been claimed that this procedure makes it possible to analyze complex systems and problems in a less inductive manner. GERT is primarily an analytical technique, but we will also be discussing a number of simulation versions of the GERT system.

GERT (GRAPHICAL EVALUATION AND REVIEW TECHNIQUE) IS A PROCEDURE WHICH COMBINES THE DISCIPLINES OF FLOWGRAPH THEORY, MOMENT GENERATING FUNCTIONS, AND PERT TO OBTAIN A SOLUTION TO STOCHASTIC PROBLEMS. IT HAS BEEN CLAIMED THAT THIS PROCEDURE MAKES IT POSSIBLE TO ANALYZE COMPLEX SYSTEMS AND PROBLEMS IN A LESS INDUCTIVE MANNER.

The GERT is made up of logical nodes and directed branches. This slide shows three input characteristics for the logical nodes and two output characteristics. The exclusive-or node will be realized as soon as one of the paths incident to the node is completed. The inclusive-or node behaves in the same manner but the exclusive-or node system is drawn in such a fashion that one and only one path incident to the node can be realized in any particular realization of the node. The AND input node behaves in the same fashion as the PERT node in that all paths incident to the AND node must be completed before any output is possible. On the output side, the DETERMINISTIC node causes all branches emanating from the node to be realized. For the PROBABILISTIC output, one and only one branch emanating from the node will be taken. From a notational standpoint, there are six possible nodes. The information on a directed branch includes the probability that a branch will be taken. This probability will be one unless it emanates from a probabilistic output side in which case the sum of the probabilities emanating from the node will add up to one. In addition, the branch includes a function of an additive parameter, usually time. The probability and the additive factor are combined into a W function which is made up of the product of the probability and the moment generating function of the additive element.

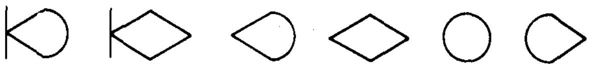
Logical nodes. A node in a stochastic network consists of an input (receiving, contributive) side and an output (emitting, distributive) side. In this chapter we will consider three logical relations on the input side and two types of relations on the output side. The three logical relations on the input side are:

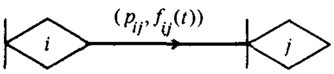
Name	*Symbol*	*Characteristic*
EXCLUSIVE-OR		The realization of any branch leading into the node causes the node to be realized; however, one and only one of the branches leading into this node can be realized at a given time.
INCLUSIVE-OR		The realization of any branch leading into the node causes the node to be realized. The time of realization is the smallest of the completion times of the activities leading into the INCLUSIVE-OR node.
AND		The node will be realized only if all the branches leading into the node are realized. The time of realization is therefore the largest of the completion times of the activities leading into the AND node.

On the output side, the two relations are defined as:

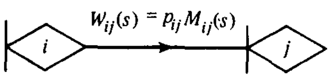
Name	*Symbol*	*Characteristic*
DETERMINISTIC		All branches emanating from the node are taken if the node is realized, i.e., all branches emanating from this node have a p parameter equal to 1.
PROBABILISTIC		Exactly one branch emanating from the node is taken if the node is realized.

For notational convenience, the input and output symbols are combined below to show that there are six possible types of nodes:



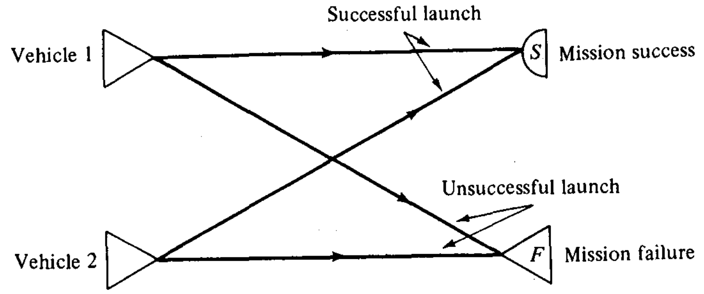


A directed branch



GERT element

This slide demonstrates the use of the various input and output node characteristics in a space rendezvous environment. In the top figure, two vehicles must be successfully launched before mission can be a success. The two outputs for the model show mission success which is only realized if both vehicles are launched successfully, thus the AND type input side, and mission failure which will be realized if either or both of the vehicle launchings are unsuccessful, thus the INCLUSIVE-OR input. The diagram on the lower portion represents an expansion of the earlier diagram and shows a modification to allow for measuring maneuverability capabilities along with successful launching. Nodes 1 and 2 will only be realized if both vehicles are launched successfully because of the AND type input side. If both vehicles are launched successfully, node S will be realized if and only if one of the vehicles is successful in maneuvering. Therefore, in this model we will have success if both vehicles are launched and at least one has maneuvering capabilities.



Stochastic network model of the rendezvous of two vehicles.

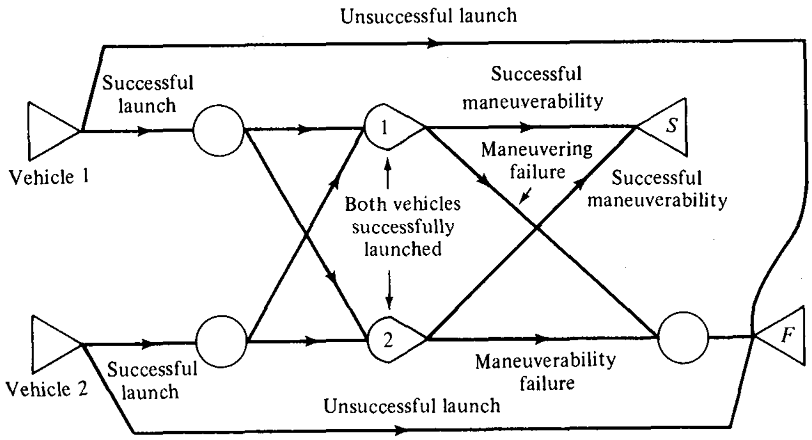
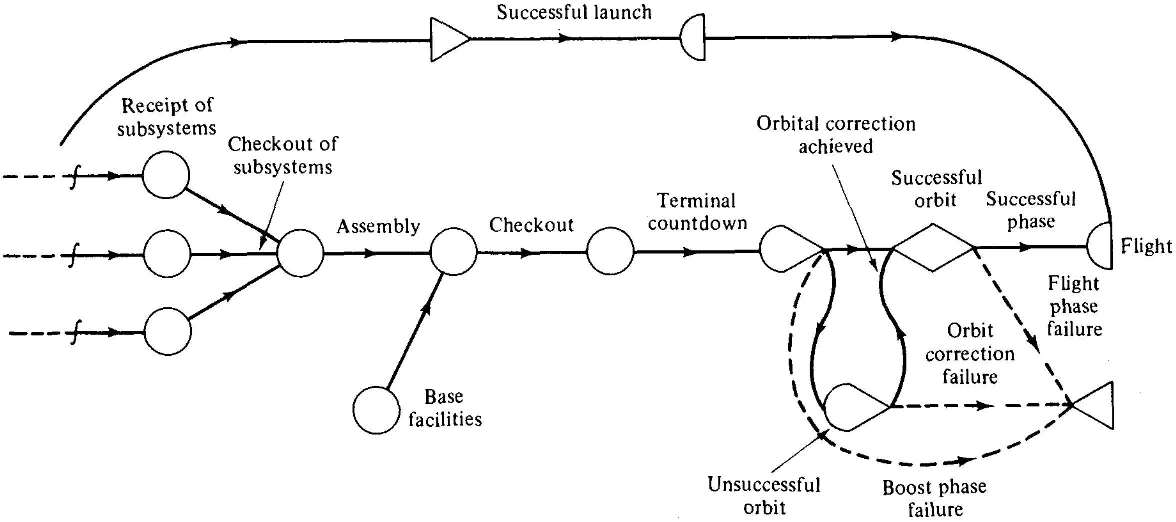


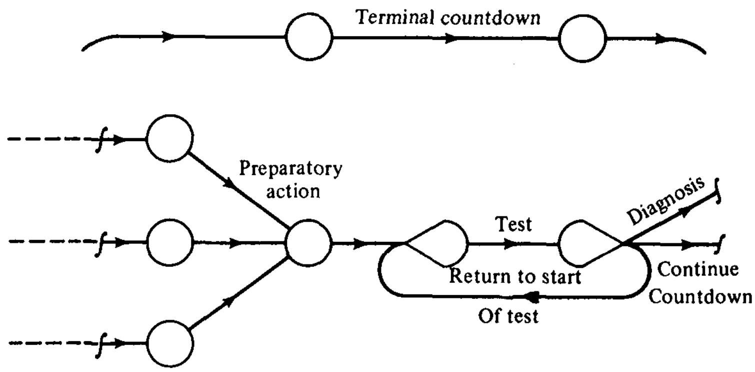
Figure modified to include maneuvering success of the rocket.

The previous slides showed networks which were highly aggregated models of complex operations. One of the advantages of stochastic networks is their usefulness at many levels within a problem area. For example, the branch "successful launch" can be divided into many branches and nodes. For example, the network shown on this slide illustrates this concept. In this network the AND node plays a predominant role in the activity up to and including the terminal countdown. This is due to the fact that all activities must be performed prior to lift-off. This, of course, is a simplified view of the system; however, it serves the purpose of illustrating that part of a stochastic network can be a PERT-type network. After the terminal countdown inclusive-or possibilities are presented and the probabilistic output nodes are shown. The event represented by the node labeled "Successful Orbit" is an EXCLUSIVE-OR node since a successful orbit can occur in two mutually exclusive ways. The broken lines represent activities that do not contribute to the successful launch but are branches associated with the system models. In this case, they would lead to the node "unsuccessful launch."



Detailed model of the successful launch branch.

Continuing this example, consider the branch "terminal countdown" a segment of which can be represented by the network shown on this slide. The network shows three preparatory actions such as power on, stimuli calibrated, and recorder-on, which are required before the test can begin. The test is performed and based on the results of the test the countdown is continued, diagnosis is initiated, or the test is performed again. This last action illustrates the concept of feedback in stochastic networks. The networks on this and previous slides are obviously not complete descriptions of countdown procedure but are useful in illustrating the communications capabilities of GERT. Also, by decomposing the problem into segments we can compute parameters of interest for the aggregate model, thus the probability of a successful launch could be computed by evaluation of a more detailed networks.



Detail model of the terminal countdown branch.

The networks made entirely of EXCLUSIVE-OR nodes form a very special class of GERT problems. These problems have been studied in great detail and they can be analytically solved using flowgraph techniques. We will concentrate on some applications of this special class of problems before moving into the simulation mode.

THE EXCLUSIVE-OR GERT NETWORKS FORM A SPECIAL CLASS

WHICH CAN BE SOLVED ANALYTICALLY USING FLOWGRAPH

TECHNIQUES.

This slide attempts to summarize the steps involved in the solutions of problems involving EXCLUSIVE-OR GERT networks. First, the analyst converts a qualitative description of the system into the network form and then he collects the necessary data to describe the elements of the network. He next calculates the topology equation to determine the equivalent function for solving the network. The formula shown on this slide is a vehicle for evaluating flowgraphs, a technique used in feedback control systems. After the W function has been calculated, it is possible to extract specific pieces of information such as the probability associated with W function and the moment generating function of the additive parameter associated with the elements analyzed. From this moment generating function moments such as the mean, variance, etc., can be calculated. Finally, as the last step inferences are drawn from the analysis and modifications are made.

STEPS ON THE SOLUTION OF PROBLEMS BY MEANS OF GERT

(FOR "EXCLUSIVE-OR" NETWORKS)

1. CONVERT A QUALITATIVE DESCRIPTION OF A SYSTEM OR PROBLEM TO A MODEL IN NETWORK FORM.

2. COLLECT NECESSARY DATA TO DESCRIBE THE TRANSMITTANCES OF THE NETWORK.

3. APPLY THE TOPOLOGY EQUATION TO DETERMINE THE EQUIVALENT FUNCTION OR FUNCTIONS OF THE NETWORK.

$$W_E(s) = (\ (\text{PATH X NON-TOUCHING LOOPS}))/\ \text{LOOPS}$$

$$= P_E M_E(s)$$

4. CONVERT THE EQUIVALENT FUNCTION INTO THE FOLLOWING TWO PERFORMANCE MEASURES OF THE NETWORK:

   4.1 THE PROBABILITY THAT A SPECIFIC NODE IS REALIZED.

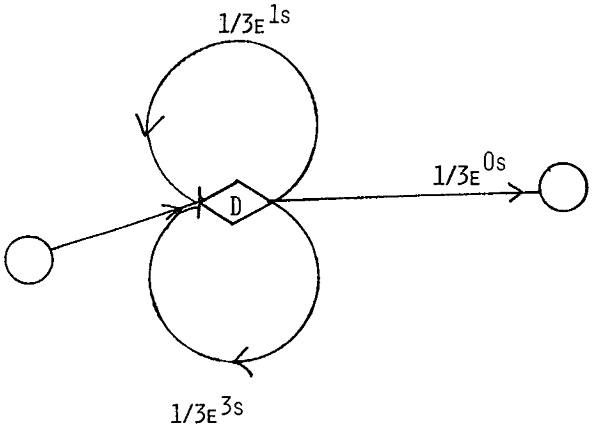
   $$P_E = W_E(s)\Big|_{s=0}$$

   4.2 THE MOMENT GENERATING FUNCTION OF THE TIME ASSOCIATED WITH A NODE IF IT IS REALIZED.

   $$M_E(s) = W_E(s)/P_E$$

5. MAKE INFERENCES CONCERNING THE SYSTEM UNDER STUDY FROM THE INFORMATION OBTAINED IN 4 ABOVE.

This slide demonstrates the use of this procedure on a rather simple network. The network represents a prisoner in the dungeon. The dungeon has three doors: one door leading to freedom, one leading to a long tunnel, and a third leading to a short tunnel. The prisoner selects a door at random and if it leads to a tunnel, he tours the tunnel eventually returning to the dungeon. Selection procedure continues until freedom is found. The network demonstrates the visual attractiveness of this technique. This problem could be solved by semi-Markovian processes which are generally difficult to understand by practitioners. The network is then converted into a W function which is the equivalent transmittance in the beginning to the end of the network. This function includes the probability of the prisoner gaining freedom and some measure of time associated with this time in the system. It is interesting to note that the moment generating function for time in the short tunnel is moment generating function of a constant 1 time unit, and the moment generating function for the long tunnel is 3 time units. The probability of exiting from the dungeon is found to be one and thus the moment generating function is equivalent to the W function. The mean time to exit this system is the first derivative of the moment generating function with respect to "s" evaluated at s=0. The mean is found to be 4 time units. The variance and other moments can be obtained from the W function.



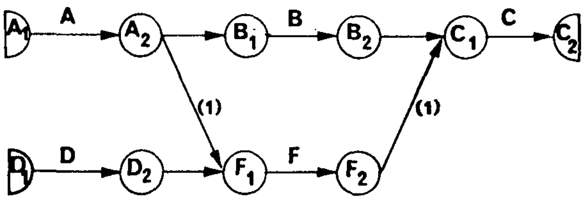
$$W_E(s) = \frac{1/3E^{s}}{1-(1/3E^{s}+1/3E^{s})}$$

$$P_E = W_E(s)\Big|_{s=0} = 1$$

$$M_E(s) = W_E(s)/P_E = W_E(s)$$

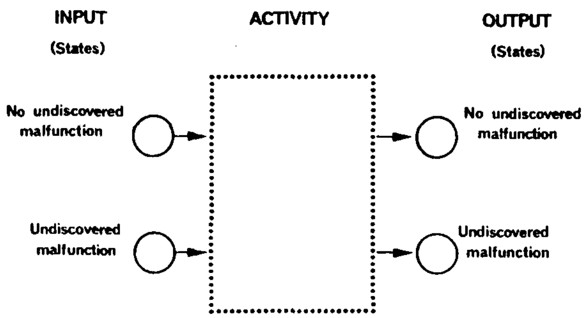
$$\mu = \frac{DM_E(s)}{DS}\Bigg|_{s=0} = \underline{4 \text{ UNITS}}$$

To continue our interest in GERT models of countdown procedures, consider this and the following slides as further application of this technique. In the figures shown on this slide, we have a PERT network description of a countdown. The five activities are A,B,C,D,E. The subscript 1 indicates a start of an activity and the subscript 2 indicates the end of an activity. Two types of relationships are used in this network: (1) functional precedence which implies an activity on the same horizontal line deals with the same equipment module and hence these activities can be used to detect malfunctions in a given module; and (2) time precedence relationships.



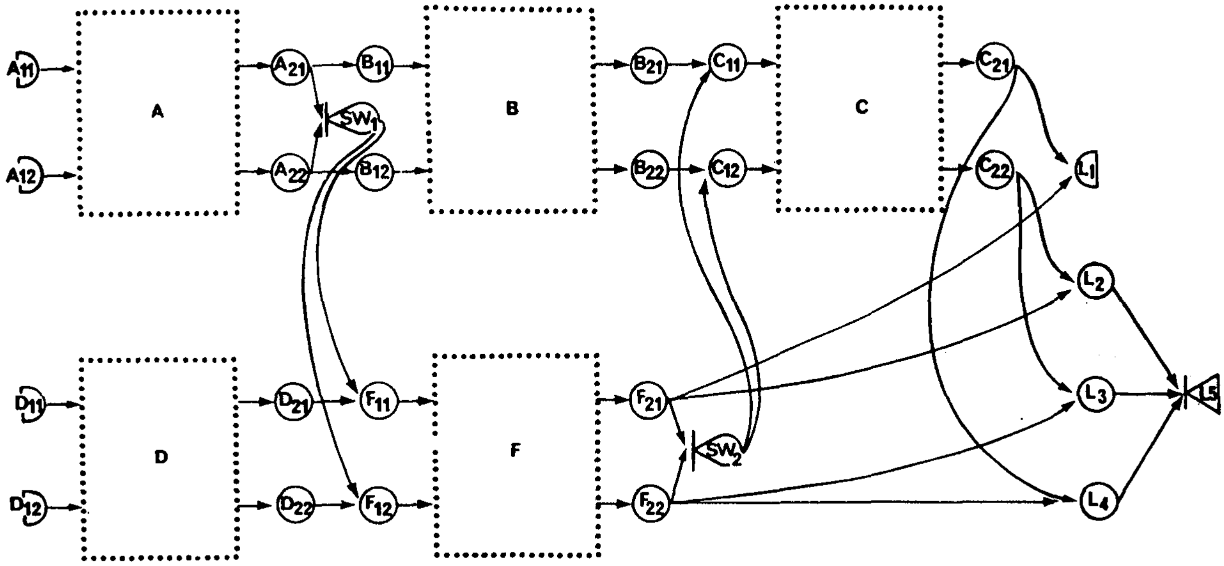
PERT network description of a countdown

This slide gives a two-port description of an activity. Inputs are "no undiscovered malfunction" and "undiscovered malfunction." The outputs are "no undiscovered malfunction," and "undiscovered malfunction."



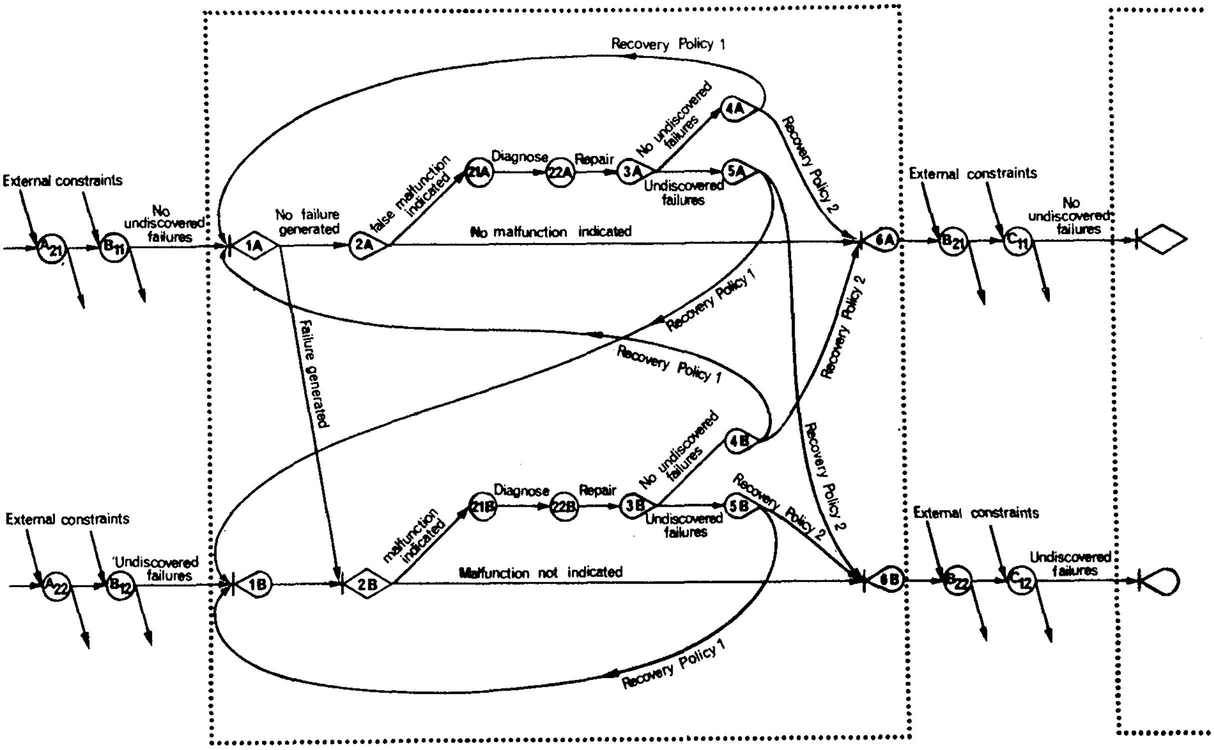
A two-port description of an activity

This slide expands the countdown representation shown using the two-port representation used in the previous slide. In this figure, node L_1 indicates a launch with no undiscovered malfunctions, and node 5 indicates a launch with undiscovered malfunctions. The performance measure of interest is the probability of obtaining successful launch which is the function of the probability of realizing node L_2 and the probability that the time to achieve node L_1 is within a specified launch window. To obtain this performance measure, first an analytical determination of two-port as detailed is made. Then the probability and times as analytically determined were used in a simulation model.



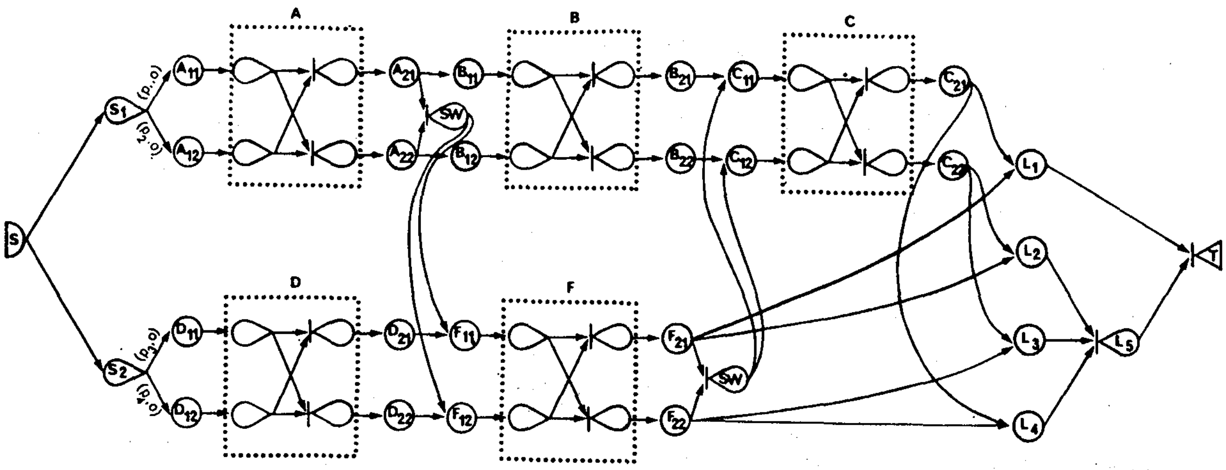
Expanded network description of a countdown

This slide represents an analytical determination of the two-port phenomenon described, and would be incorporated in the detailed complete network shown in the next slide.



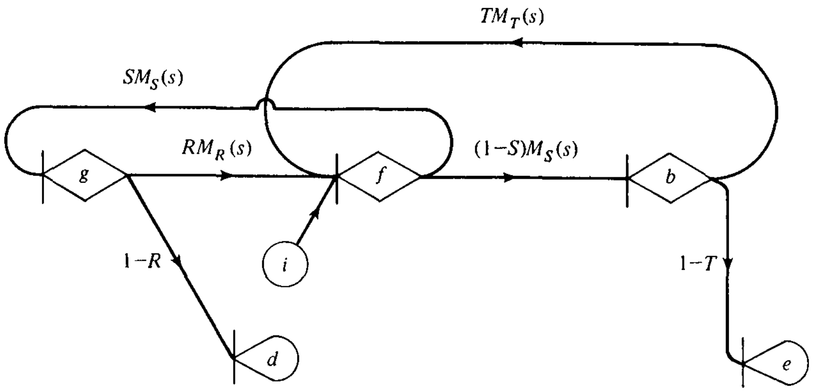
Detailed description of an activity

Finally, this slide shows the GERT network description of the network countdown. Questions regarding the work activities reflected by the performance measure can then be proposed and modifications of the model can be achieved.



GERT network of a countdown
(Note: Except as noted all unmarked branches not enclosed in boxes have probability = 1, time = 0.

This slide demonstrates information available from the GERT network. For systems that involve expensive parts, attempts are often made to salvage or recover parts for future use. For such systems, we might like to determine the expected life of one of these parts in terms of both time and number of users. This model considers the possibility of a recovery of a rocket after a test, assuming that there is no deterioration in the system. Node 1 represents the initial flight, node F represents any flight, G is a successful flight, B an unsuccessful flight, D non-recovery after G, and E non-recovery after B. Information included on the diagram is the probability of successful flight and appropriate probabilities of recovery. Time parameters include flight time and various recovery times.



GERT representation of a rocket recovery problem.

Events:

i = initial flight.
f = any flight.
g = a successful flight.
b = an unsuccessful flight.
d = nonrecovery after g.
e = nonrecovery after b.

Probabilities:

S = probability of a successful flight.
R = probability of a recovery after g.
T = probability of a recovery after b.

MGF:

$M_S(s)$ = MGF of a flight time.
$M_R(s)$ = MGF of the time of recovery after g.
$M_T(s)$ = MGF of the time of recovery after b.

G. E. Whitehouse, Systems Analysis and Design Using Network Techniques, c. 1973, pp. 382-383. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, NJ.

Typical information that I was able to calculate for this model of the recoverable rocket using techniques reported in reference 1 includes the life of the rocket, the number of flights until non-recovery, the probability of at least one successful flight, and the first passage time until we achieved our first success. There are a number of other pieces of information which could be analytically derived from models of this sort. These all represent viable performance measures in this type of problem.

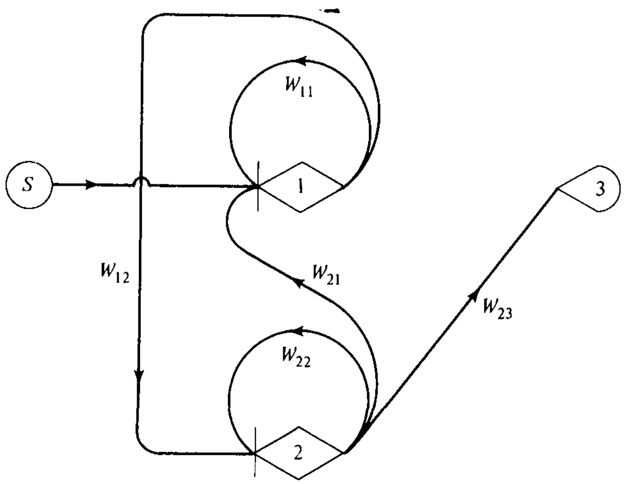
TYPICAL INFORMATION AVAILABLE FROM THE GERT MODEL OF THE RECOVERABLE ROCKET

1. THE LIFE OF THE ROCKET
2. THE NUMBER OF FLIGHTS UNTIL NONRECOVERY
3. PROBABILITY OF AT LEAST ONE SUCCESS
4. THE FIRST PASSAGE TIME TO A SUCCESS

The network in this system represents a model of a repair situation. Suppose a device is being developed for a given application. The application is such that the device when put into operation either succeeds or fails to accomplish what it is designed to do. Suppose further there is only one thing that can go wrong with the device and the device will eventually fail due to this fault. The whole purpose of the development effort on the device is to discover what the cause of the failure is and then attempt to redesign or fix the device so it will not fail again. Assume repair either fixes the device or not; i.e., the probability "1-p" of defective operation is constant until the device is completely fixed and always works. The development effort then consists of repeated trials on the device. If the device operates successfully on any given trial, the designer or development engineer decides to make no redesign action. He proceeds to the next trial on the chance that he has already fixed the device and its probability of failure is zero. If it fails on any given trial, the engineer goes to work on it and has probability "a" of fixing the device permanently prior to the next trial. The network includes the moment generating functions for the repair time and trial times. There are three outcomes that are possible from a given outcome: (1) the trial is successful given a device is faulty, (2) the trial is a failure given the device is faulty, and (3) the trial is successful; the device is fixed.

These outcomes are represented by the events on the GERT network. The moment generating function of the time to successfully diagnosis and fix the fault is shown by the function at the top of the slide. The information at the bottom of the slide demonstrates the various information available by further analyzing the network and includes the number of trials necessary until the device is successfully fixed. We can also find the number of failures before we successfully complete our repair.

$$M_{S,3}(s) = \frac{a(1 - p)M_t(s)M_r(s)}{1 - pM_t(s) - (1 - a)(1 - p)M_r(s)}$$



where

$W_{11} = pM_t(s)$,
$W_{12} = (1-p)M_t(s)$,
$W_{21} = (1-a)pM_r(s)$,
$W_{22} = (1-a)(1-p)M_r(s)$

and $W_{23} = aM_r(s)$.

GERT representation of a reliability repair model.

Some examples of the information available from this model using counters are:

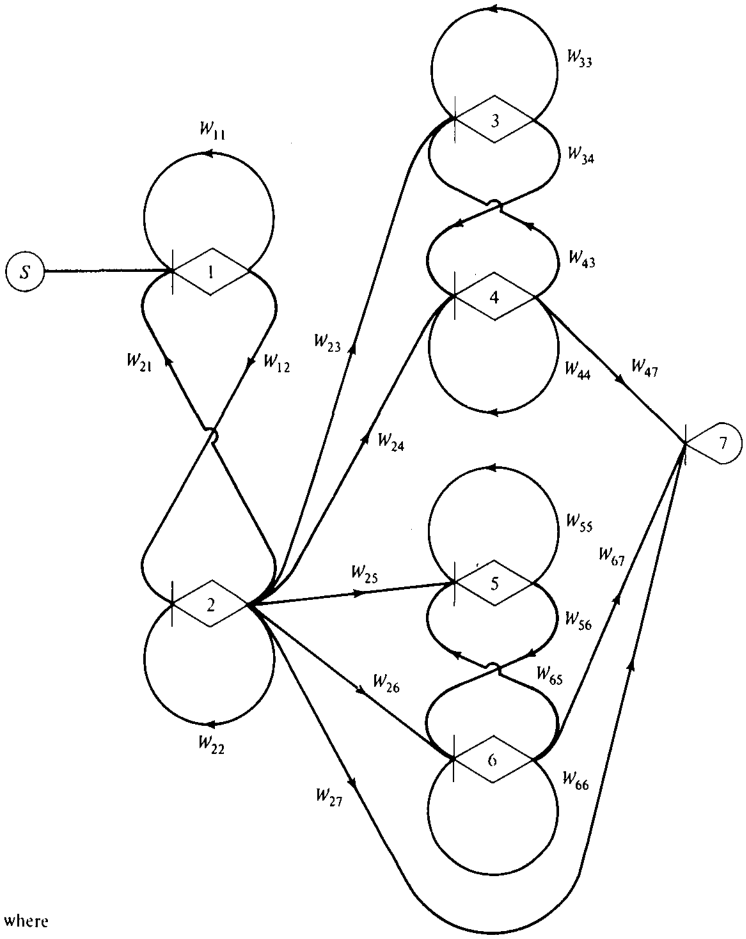
1. The MGF of the number of trials necessary until the device is operative is found by tagging all elements with an e^c tag, solving the graph for $M_{S,3}(s, c)$, and then equating $s = 0$. The MGF of the number of trials until the device is dependable is:

$$M_{S,3}(s, c)_{s=0} = \frac{a(1 - p)e^{2c}}{1 - pe^c - (1 - a)(1 - p)e^c}$$

2. The MGF of the number of failures can be investigated if we tag all the elements entering the nodes representing a failure (in this case node 2) with an e^c tag and then solve for $M_{S,3}(s, c)$:

$$M_{S,3}(s, c)|_{s=0} = \frac{a(1 - p)e^c}{1 - p - (1 - a)(1 - p)e^c}$$

The network becomes considerably more complicated when there is more than one fault. The network in this slide demonstrates a version which would involve the attempt to analyze and fix two levels of faults.



where

$W_{11} = \bar{p}_1 \cap \bar{p}_2 M_t(s)$,
$W_{12} = p_1 \cup p_2 M_t(s)$,
$W_{21} = \bar{a}_1 \cap \bar{a}_2 \cap \bar{p}_1 \cap \bar{p}_2 M_r(s)$,
$W_{22} = \bar{a}_1 \cap \bar{a}_2 \cap p_1 \cup p_2 M_r(s)$,
$W_{23} = a_1 \cap \bar{a}_2 \cap \bar{p}_2 M_r(s)$,
$W_{24} = a_1 \cap \bar{a}_2 \cap p_2 M_r(s)$,
$W_{25} = \bar{a}_1 \cap a_2 \cap \bar{p}_1 M_r(s)$,
$W_{26} = \bar{a}_1 \cap a_2 \cap p_1 M_r(s)$,
$W_{27} = a_1 \cap a_2 M_r(s)$,
$W_{33} = \bar{p}_2 M_t(s)$,
$W_{34} = p_2 M_t(s)$,
$W_{43} = \bar{a}_2 \cap \bar{p}_2 M_r(s)$,
$W_{44} = \bar{a}_2 \cap p_2 M_r(s)$,
$W_{47} = a_2 M_r(s)$,
$W_{55} = \bar{p}_1 M_t(s)$,
$W_{56} = p_1 M_t(s)$,
$W_{65} = \bar{a}_1 \cap \bar{p}_1 M_r(s)$,
$W_{66} = \bar{a}_1 \cap p_1 M_r(s)$,
$W_{67} = a_1 M_r(s)$,
$\bar{a}_i = 1 - a_i$,
and $\bar{p}_i = 1 - p_i$.

GERT representation of a two-level reliability repair model.

Events:

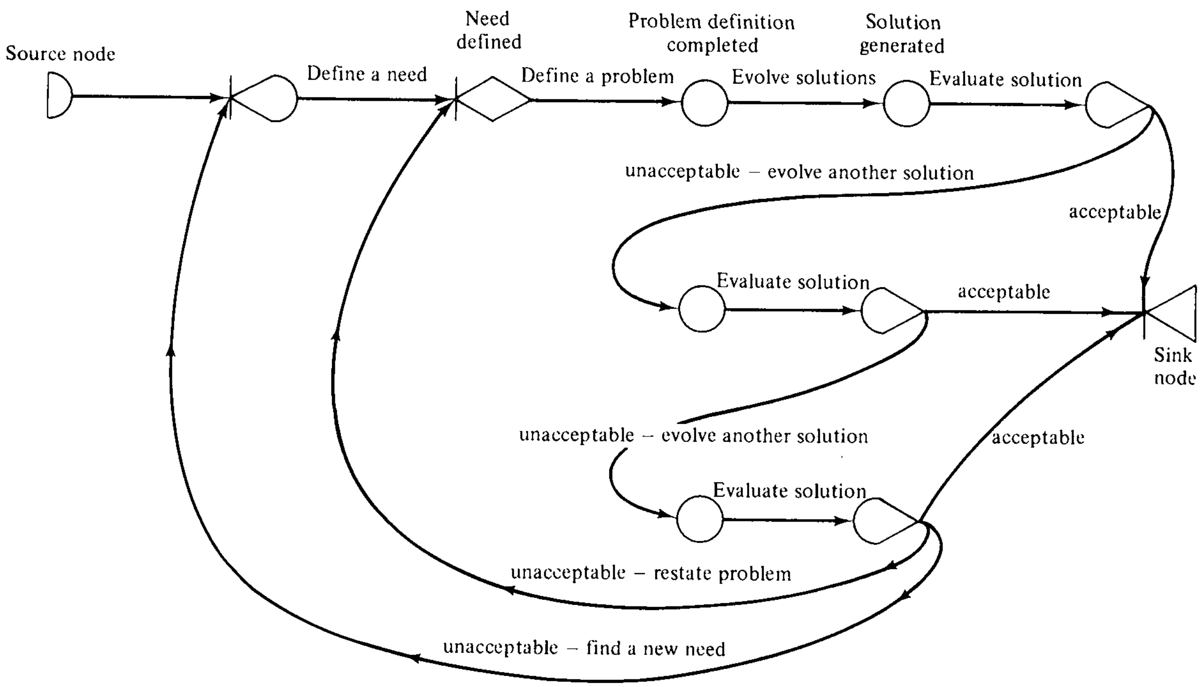
1 = success, given that both modes broken.
2 = failure, given that both modes broken.
3 = success, given that mode 1 fixed.
4 = failure, given that mode 1 fixed.
5 = success, given that mode 2 fixed.
6 = failure, given that mode 2 fixed.
7 = success, given that both modes fixed.

Probabilities:

p_1 = probability of mode 1 causing failure.
p_2 = probability of mode 2 causing failure.
a_1 = probability of fixing mode 1.
a_2 = probability of fixing mode 2.

G. E. Whitehouse, Systems Analysis and Design Using Network Techniques, c. 1973, pp. 386-387. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, NJ.

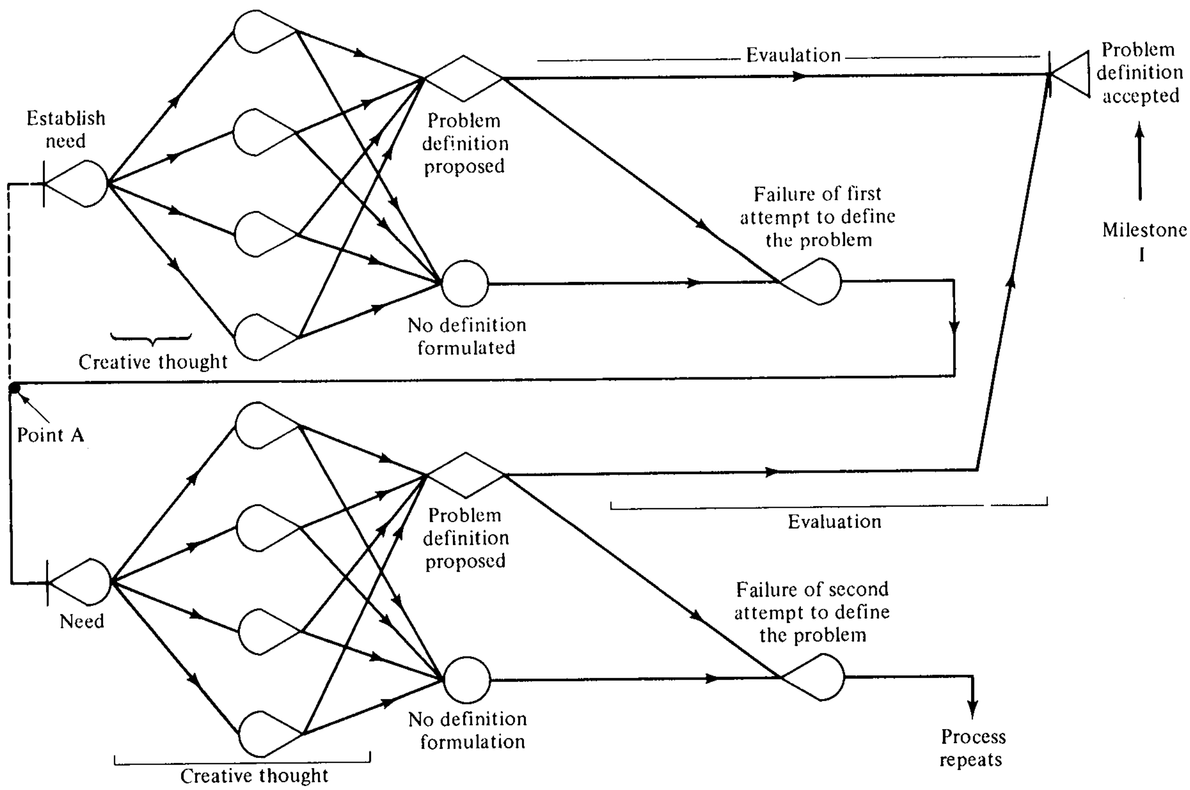
This slide attempts to show the use of GERT modeling in the research and development situation. Basically, the research and development process consists of five milestones: (1) the completion of model definition, (2) the completion of research activities, (3) acceptance of the proposed solution, (4) completion of the prototype, and (5) implementation of the solution. The network on this slide represents the general network model of the activities of achieving the first three milestones. Since a hierarchical network development procedure will be used, the activities are defined in broad terms. The network in this slide illustrates three attempts at obtaining solutions for a given problem. If all three solutions are unacceptable, then either a redefinition of the problem will be made, or a new need will be explored and the researcher will essentially give up on the previous problem.



GERT research model.

G. E. Whitehouse, Systems Analysis and Design Using Network Techniques, c. 1973, p. 410. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, NJ.

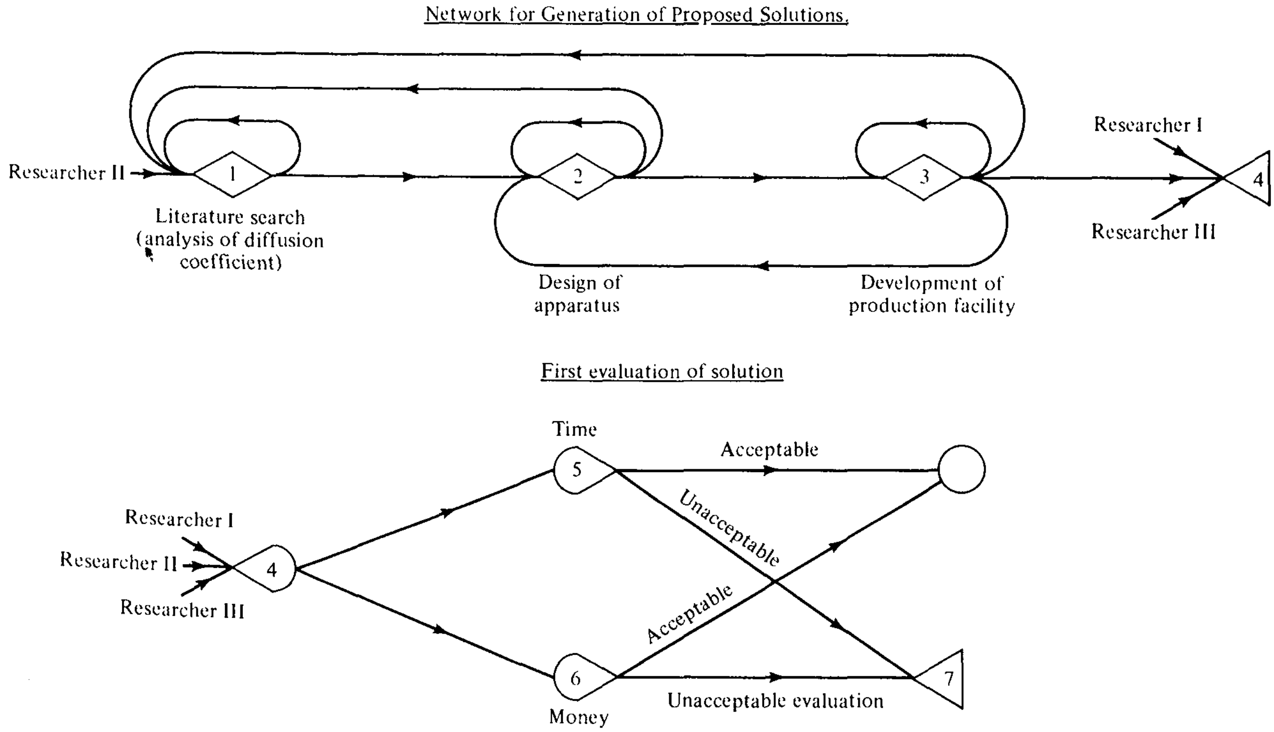
The network in this slide represents the activities in problem definition. On the chart is shown a creative thought process following the establishment of the need. As shown, there are four separate efforts involved in attempting to defined the problem. On the outside of the node following the creative thought, a probabilitic node is used to indicate that the problem is either defined or not based upon the creative thought efforts. No definition will evolve only if all four efforts are unsuccessful. The point A in this diagram represents a possible regeneration point of the problem definition process. If the characteristics of the activities involved in a problem definition do not change based upon previous attempts at problem definition, a return to the original start node can be made and the network need not be repeated as shown on the bottom half of this slide. Since learning occurs in the research and development process, it is more reasonable to indicate a repeat with new parameters of activities involved in the problem definition. This lack of regeneration points in the research and development process probably, I believe, has hindered many analysis attempts.



GERT problem definition.

G. E. Whitehouse, Systems Analysis and Design Using Network Techniques, c. 1973, p. 411. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, NJ.

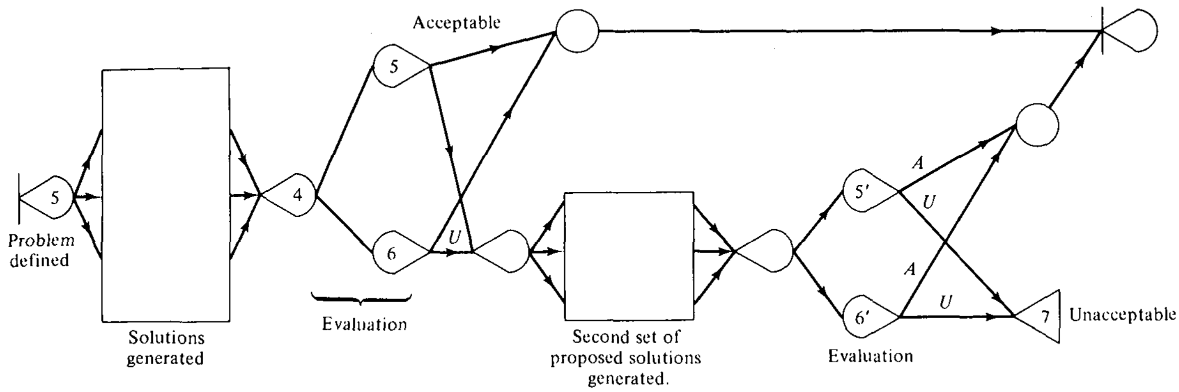
The networks in this slide demonstrate one possible representation of the research activity for one researcher involved in proposing solutions. Also shown in this slide is the evaluation procedure model in network form for considering both time and cost considerations involved in proposed solutions. Thus, a model will only be successful if both time and money are considered within acceptable bounds.



Network for generation of proposed solutions.

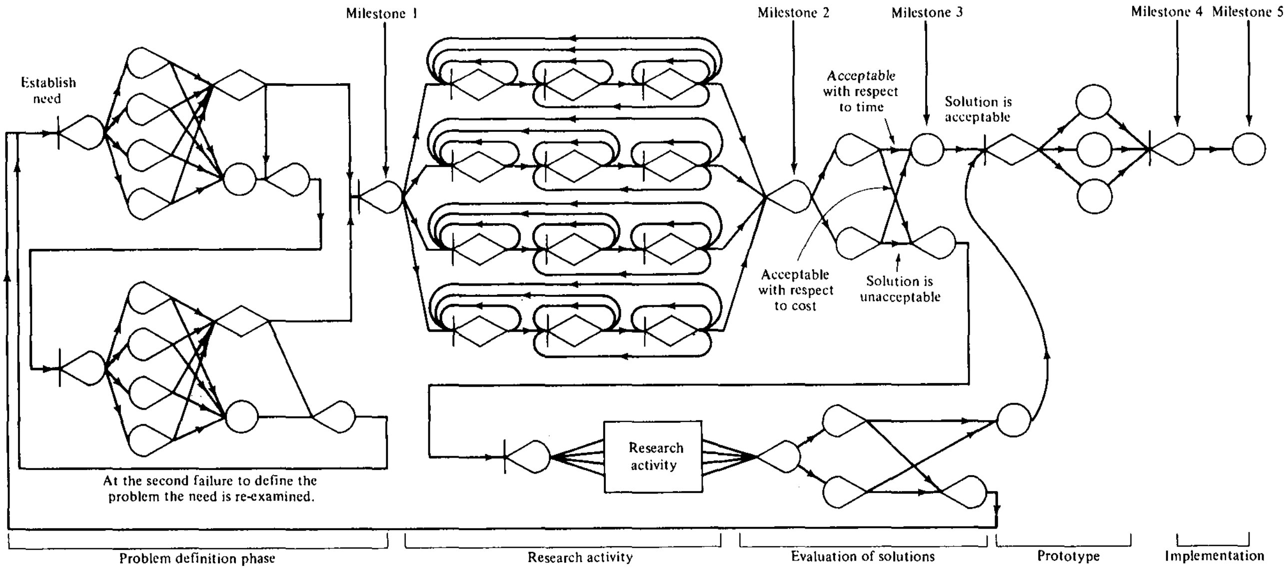
G. E. Whitehouse, Systems Analysis and Design Using Network Techniques, c. 1973, p. 413. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, NJ.

The figure in this slide shows a network for generating and evaluating the solutions serially. Thus, this figure includes milestones 1 and 2.



Series proposals.

Tying the detailed elements together results in the network shown in this slide which illustrates one possible network for representing the scientific method approach to planning research and development. This also demonstrates how large scale GERT networks can be developed in a modular sense.



A GERT network representation of an example R & D project.

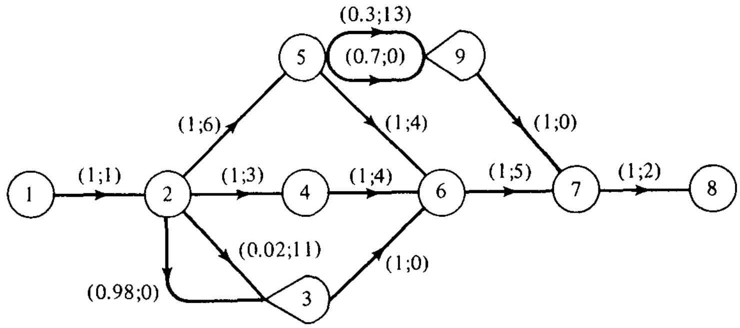
G. E. Whitehouse, Systems Analysis and Design Using Network Techniques, c. 1973, pp. 414-415. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, NJ.

There have been many applications of GERT models in many areas. This slide lists a few which might be of interest to the NASA environment. They include rather elaborate project management models along with quality control and reliability models. Inventory and production systems have been modeled as have been manufacturing and queueing systems. Communications systems have been modeled for the government, along with models for fault detection in electronic circuitry.

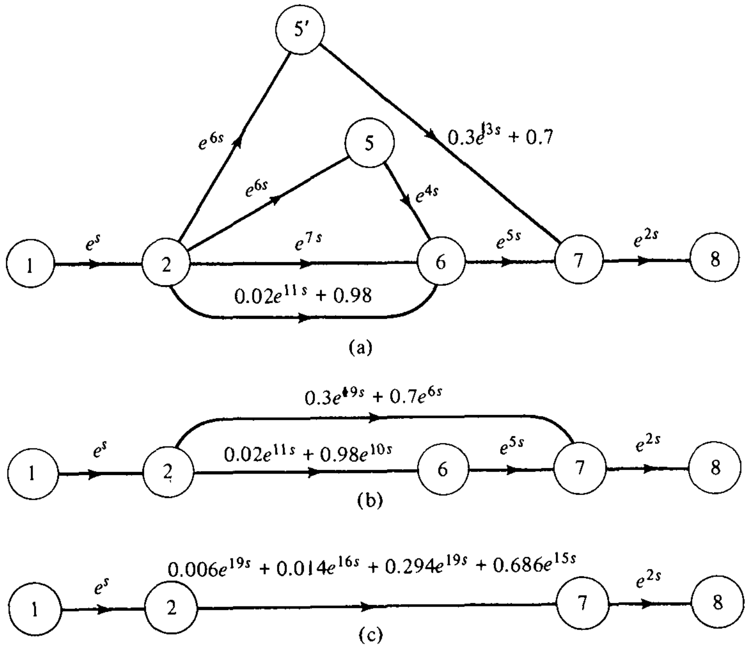
OTHER AREAS OF APPLICATION

1. PROJECT MANAGEMENT
2. QUALITY CONTROL COST MODELS
3. INVENTORY MODELS
4. PRODUCTION MODELS
5. MANUFACTURING SYSTEMS
6. QUEUEING ANALYSIS
7. COMMUNICATION SYSTEMS
8. FAULT DETECTION

It would be nice to be able to model all nodes and solve them analytically. Work has not developed along these lines. Most work has attempted to take the AND and INCLUSIVE-OR nodes and convert them into EXCLUSIVE-OR nodes. This slide demonstrates some alternatives and the conversion actually gives an incorrect analysis to the model in question. Most efforts for the solution of more complicated networks involving nodes other than the EXCLUSIVE-OR nodes are now solved using simulation techniques.

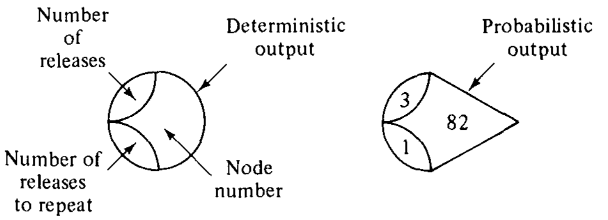


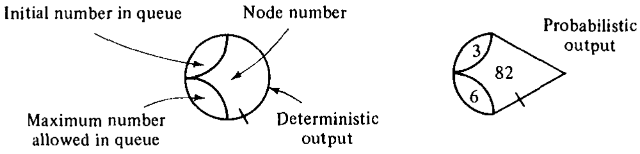
Representation of a network composed of AND nodes as one with EXCLUSIVE-OR nodes.



Reduction of a network composed of AND nodes.

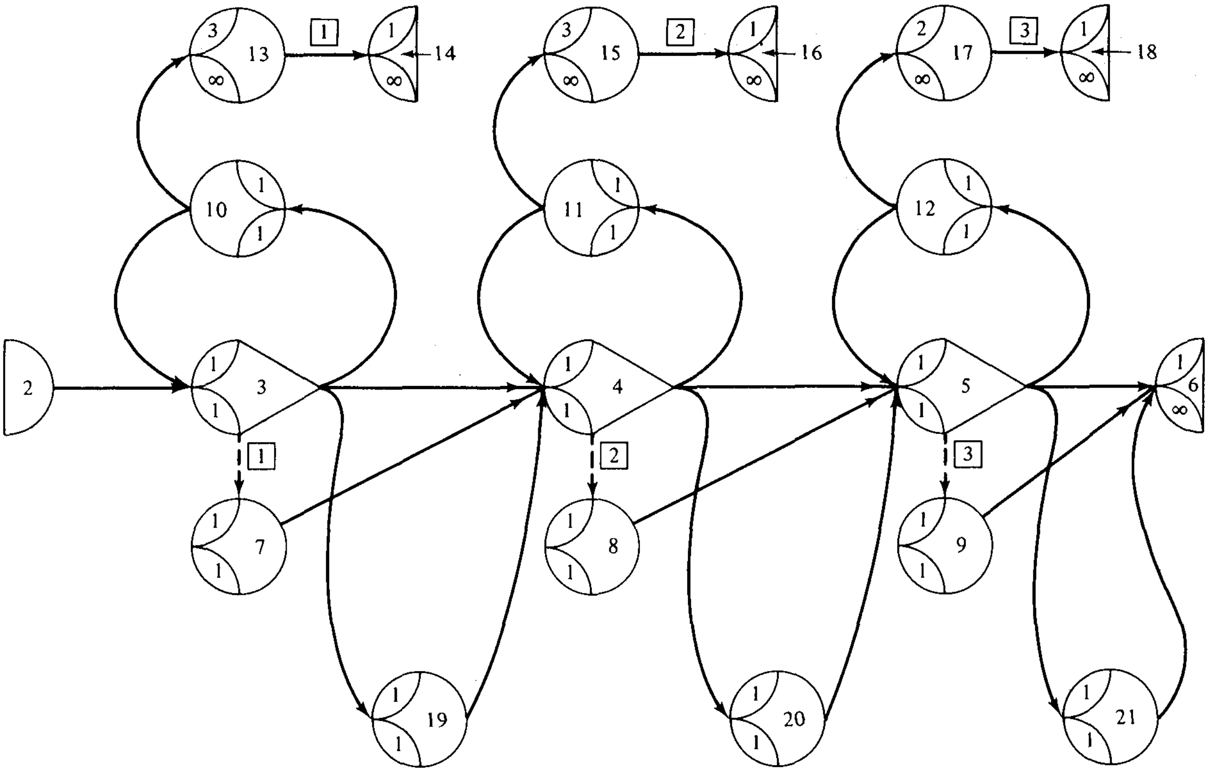
This slide represents the nodes used in the so-called GERTS IIQ Model. These nodes have DETERMINISTIC and PROBABILISTIC output in addition to having the capability of modeling INCLUSIVE-OR, EXCLUSIVE-OR and AND types of inputs. The input side is released based upon the information shown on the node in question. In addition to these nodes, there are also nodes involved in queueing. Nodes at the bottom of this slide show queueing type nodes. These nodes include information about the initial number in a queue, along with the maximum number acceptable in a queue. PROBABILISTIC and DETERMINISTIC outputs are possible.





Queue nodes

This slide demonstrates the use of GERT IIIQ in the analysis of experiments performed in space by a space crew. The performance of experiments in space made by a space crew is almost always severely constrained by time. Many experiments are usually proposed by a scientific community and of those proposed a subset must be chosen for a given space mission. The sequencing of those experiments which can be completed is then accomplished. A GERTS model of the sequence of experiments is developed and shown on this slide. I have assumed that the three possible outcomes from the performance of each experiment are (1) successful completion, (2) failure and (3) inconclusive results. If an experiment is successfully completed, the next experiment in the sequence is performed. If failure occurs, the experiment is scrubbed and the next experiment is performed. If the results of the experiment are inconclusive, an experiment is repeated "n" times until a success or failure occurs. The experiment is scrubbed if tried "n" times and results are still inconclusive. The network shown in this slide represents a three experiment program. Node 2 is the start node and indicates a transfer to node 3, which represents the decision point for the first experiment. If the first experiment is successful, the activity from node 3 to node 4 is realized. If the first experiment fails, the activity from node 3 to 19 is taken. The second experiment is started by transferring from node 19 to 4. If the results from the first experiment are inconclusive, the activity from node 3 to node 10 is taken. The output of node 10 is deterministic and hence when the first experiment is performed again a signal is sent to node 13 to indicate the first experiment has been performed once. Thus, for each experiment we will either transfer to node 4 or 13. When node 13 is realized, 3 times the activity from node 13 to node 14 is realized. This activity is labeled as activity 1. It causes the network to be modified by replacing node 3 by node 7. When this occurs, transfer will automatically be made to node 4. The other experiments are handled in essentially the same manner.



GERT network for the analysis and sequencing of space experiments.

EXPERIMENT CHARACTERISTICS

Experiment	*Probability of Success*	*Probability of Failure*	*Probability of Inconclusive Results*	*Allowable Numbers of Repeats*
1	0.6	0.1	0.3	3.
2	0.5	0.1	0.4	3
3	0.7	0.1	0.2	2

Experiment	*Mean Time*	*Minimum Time*	*Maximum Time*	*Standard Deviation*
1	10.0	5.0	20.0	2.0
2	20.0	15.0	25.0	1.0
3	15.0	10.0	30.0	3.0

This slide shows the computer input for these nodes. The system is a data driven system.

INPUT DATA FOR EXAMPLE

SPACE EXPS 2 5201973 400 4 40 1267	1	EX 2 10
21 1 1 1 10 1	2	EX 2 20
2 1 0 D		EX 2 30
3 3 1 1P 5 1 A		EX 2 40
4 3 1 1P 25 3 A		EX 2 50
5 3 1 1P 37 3 A		EX 2 60
6 2 1 D 37 3 A		EX 2 70
7 1 1D		EX 2 80
8 1 1D		EX 2 90
9 1 1D		EX 2 100
10 1 1D	3	EX 2 110
11 1 1D		EX 2 120
12 1 1D		EX 2 130
13 3 D		EX 2 140
14 3 1 D 1 1 F		EX 2 150
15 3 D		EX 2 160
16 3 1 D 65 1 F		EX 2 170
17 2 D		EX 2 180
18 3 1 D 50 2 F		EX 2 190
19 3 1 1D 1 1 A		EX 2 200
20 3 1 1D 27 2 A		EX 2 210
21 3 1 1D 41 3 A		EX 2 220
0		EX 2 230
10 5 20 2		EX 2 240
20 15 25 1	4	EX 2 250
15 10 30 3		EX 2 260
0		EX 2 270
1 2 3 1 2		EX 2 280
6 3 4 2 2		EX 2 290
3 3 10 4 1		EX 2 300
1 3 19 4 1		EX 2 310
5 4 5 3 2		EX 2 320
4 4 11 4 1		EX 2 330
1 4 20 4 1		EX 2 340
7 5 6 4 1		EX 2 350
2 5 12 4 1		EX 2 360
1 5 21 4 1		EX 2 370
1 7 4 2 2		EX 2 380
1 8 5 3 2	5	EX 2 390
1 9 6 4 1		EX 2 400
1 10 3 1 2		EX 2 410
1 10 13 4 1		EX 2 420
1 11 4 2 2		EX 2 430
1 11 15 4 1		EX 2 440
1 12 5 3 2		EX 2 450
1 12 17 4 1		EX 2 460
1 13 14 4 1 1		EX 2 470
1 15 16 4 1 2		EX 2 480
1 17 18 4 1 3		EX 2 490
1 19 4 2 2		EX 2 500
1 20 5 3 2		EX 2 510
1 21 6 4 1		EX 2 520
0		EX 2 530
1 3 7		EX 2 540
2 4 8	6	EX 2 550
3 5 9		EX 2 560
0		EX 2 570

The output on the slide shows the probabilities of a success, failure and inconclusive results for all experiments along with the information regarding experimental time. To reach node 4 it took over 46 time units with a standard deviation of 18 time units. In some cases, it took as little as 24 time units and in other cases took over 114 time units. The number of times experiment 1 was completed within a given time interval is presented in the histogram shown below. Similar statistical quantities are available on the other nodes in the system. Another interesting feature that we could have incorporated in this model would be to sequence experiments dependent upon the results of some other experiments.

GERTS III SUMMARY REPORT

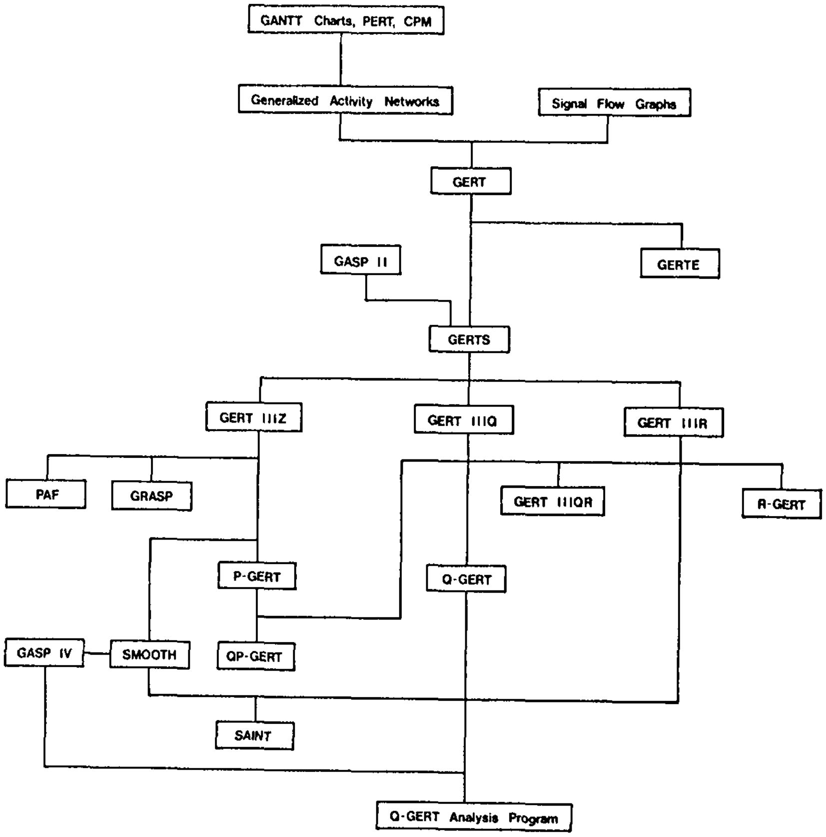
GERT SIMULATION PROJECT 2 BY SPACE EXPS
DATE 5/20/1973
"FINAL RESULTS FOR 400 SIMULATIONS"

NODE	PROB./COUNT	MEAN	STD. DEV.	# OF OBS.	MIN.	MAX.	NODE TYPE
6	1.0000	66.1562	21.8904	400.	36.7050	137.6164	A
21	0.1300	68.2321	21.8631	52.	37.9771	118.8453	A
20	0.1525	44.6018	16.2177	61.	25.6203	96.1071	A
19	0.1575	13.3597	6.6827	63.	5.5631	32.2204	A
18	0.0350	77.9196	22.1003	14.	52.1680	119.6703	F
16	0.0675	73.4611	7.2378	27.	66.5126	93.7269	F
14	0.0350	29.9889	3.0599	14.	25.3156	36.5681	F
5	1.0000	65.8379	21.5760	493.	36.7050	137.6164	A
4	1.0000	46.1740	18.7907	670.	24.5228	114.1765	A
3	1.0000	14.1291	7.5545	576.	5.0000	43.2401	A

"HISTOGRAMS"

NODE	LOWER LIMIT	CELL WIDTH	FREQUENCIES										
6	37.00	3.00	2	10	32	44	30	18	17	11	28	25	21
			17	18	12	11	14	..19	13	7	4	5	2
			6	5	4	8	1	5	3	3	0	5	
21	41.00	3.00	4	5	1	5	2	2	0	5	2	0	3
			5	2	1	1	0	4	2	1	3	0	1
			0	1	0	0	2	0	0	0	0	0	
20	27.00	2.00	4	5	7	4	5	1	2	2	3	3	2
			2	2	4	2	1	0	0	0	3	3	0
			2	0	2	0	0	0	1	0	0	1	
19	1.00	1.00	0	0	0	0	0	2	3	4	7	8	10
			9	2	0	1	2	0	0	0	5	0	0
			2	3	0	0	0	2	0	1	0	2	
18	50.00	2.00	0	0	1	0	1	1	2	0	2	0	1
			0	0	0	0	0	0	0	1	0	1	1
			0	0	0	0	1	0	0	0	0	2	
16	65.00	1.00	0	0	4	1	1	2	6	2	4	1	0
			0	0	0	1	0	1	1	0	0	1	0
			0	0	0	0	0	0	1	1	0	0	
14	1.00	1.00	0	0	0	0	0	0	0	0	0	0	0
			0	0	0	0	0	0	0	0	0	0	0
			0	0	0	1	0	1	7	0	1	4	
5	37.00	3.00	2	12	39	49	41	24	21	15	36	35	27
			22	22	13	13	17	20	13	9	5	5	3
			10	5	6	9	1	6	4	4	0	5	
4	25.00	3.00	2	59	135	73	17	31	28	30	44	55	35
			8	20	15	15	14	23	10	4	8	8	4
			10	10	2	1	6	1	0	0	2	0	
3	5.00	1.00	0	11	20	41	59	74	81	55	37	13	11
			8	10	8	11	20	13	19	15	10	10	2
			1	4	10	3	2	3	5	2	3	15	

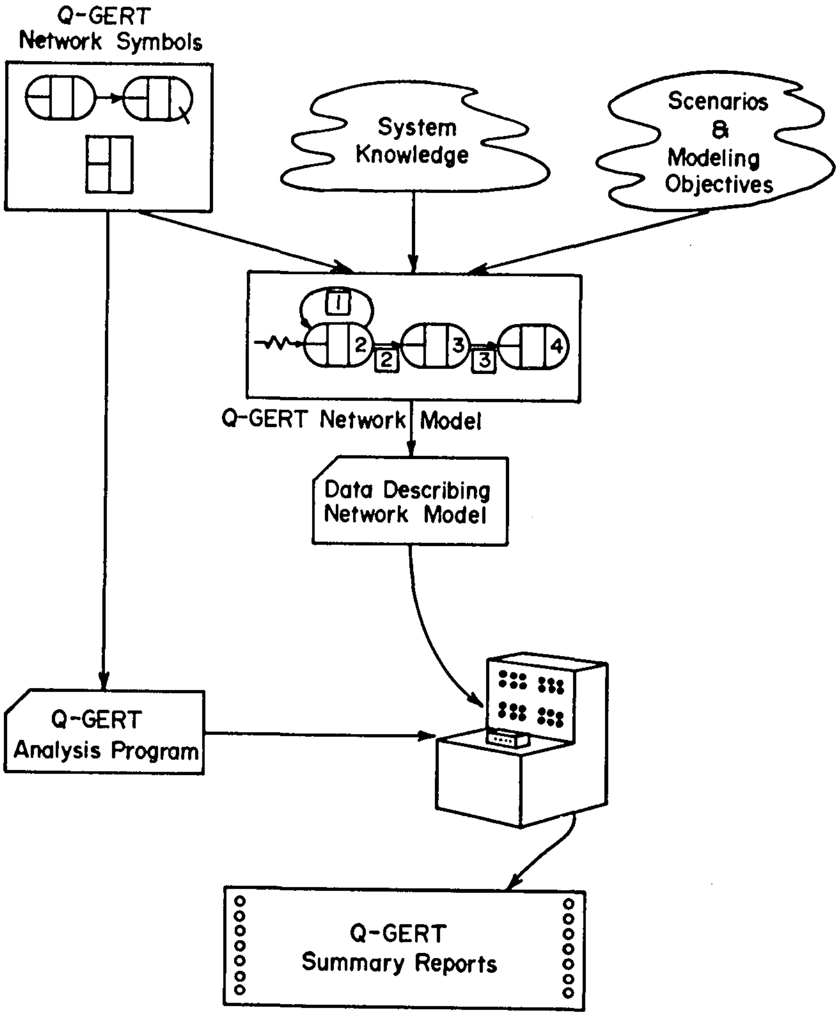
This slide shows a tree developed by Dr. Pritsker showing the evolution of various GERT techniques through the QGERT program which was developed around 1974. As you can see, there are many versions of GERT, many of which have been developed for special purposes. The diagram shows the evolution of GERT from PERT, generalized activity networks and flowgraphs, then shows the influence of simulation through such languages as GASP through various versions of GERTS III. The most significant development after GERTS III was Q-GERT.



The GERT family tree.

(from ref. 2)

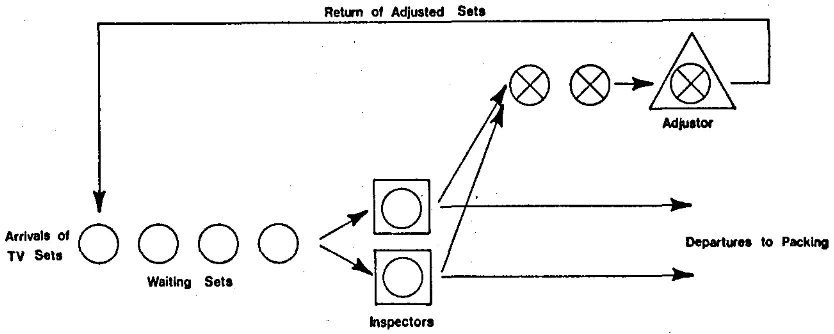
This slide shows the philosophy in developing a Q-GERT model. The network is developed in a symbolic form and is represented by the Q-GERT network model. This model is then translated into data format fed into the computer along with the Q-GERT analysis program, and typical Q-GERT summary reports are generated. This is essentially the same approach used in GERTS III, except that the Q-GERT model is much more general.



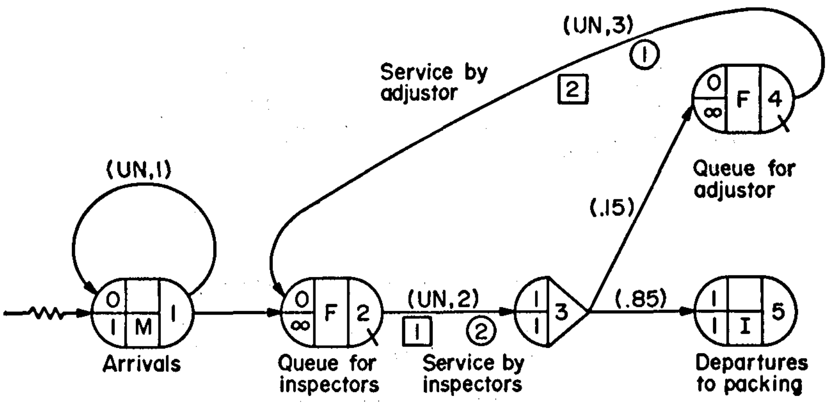
Components of Q-GERT modeling and analysis.

(from ref. 2)

Many of the models used in Q-GERT are queueing type models. The one shown on this slide is very simple and represents the inspection of TV sets. Node 1 represents the arrivals of units to the system. They then move into a queue and wait in front of two inspectors. At node 3 inspection is completed and 85 percent are packaged while 15 percent are sent to an adjustor represented by node 4. The adjustor completes his work and the TV set is returned to the inspection facility. There are numerous QGERT models, some of which are very complicated; I do not intend to spend much time on these because they are essentially the same as the network versions of the SLAM language which we will now discuss.



Schematic diagram of inspection and adjustment stations, Example 1.



Q-GERT model of inspection and adjustment stations, Example 1.

(from ref. 2)

The latest development is the SLAM simulation language. This language involves a network simulation at a higher level than Q-GERT, discrete simulation or continuous simulation. This language can combine all of these features and an analyst can use whichever type of simulation that seems to be appropriate to the model at hand. I am personally inclined to do most of my modeling in the network simulation mode, but find the ability to interact with the discrete and continuous modes a very attractive feature.

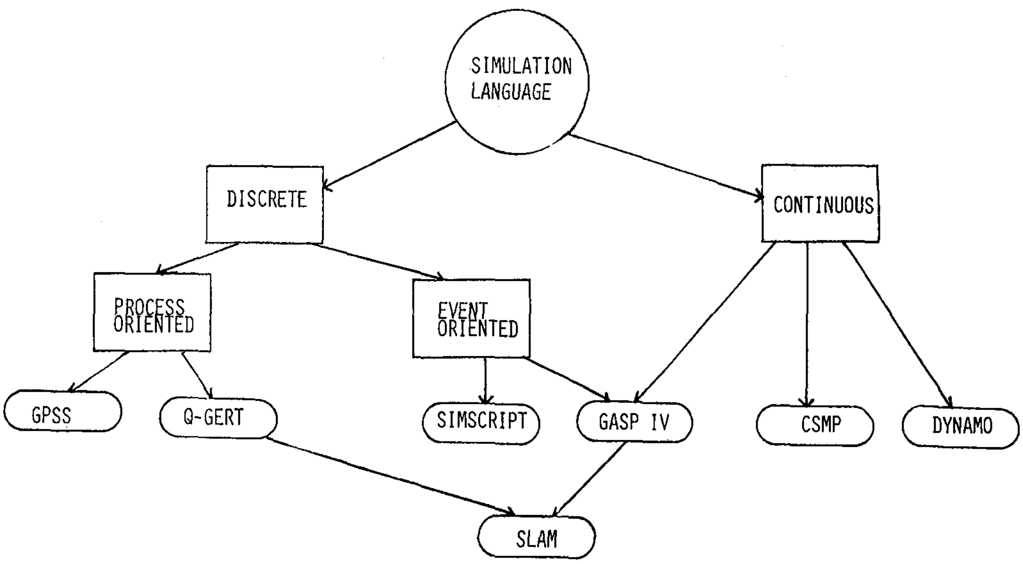
SLAM LANGUAGE

COMBINES:

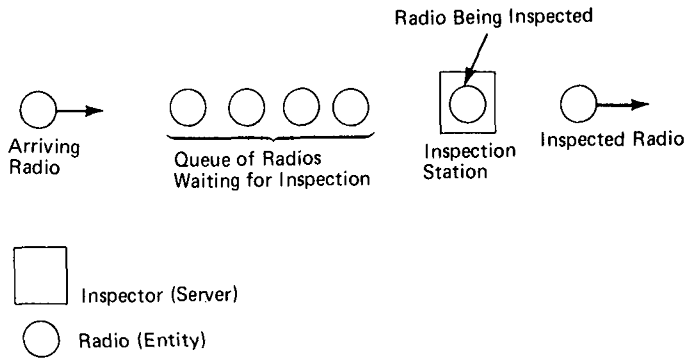
- NETWORK SIMULATION
- DISCRETE SIMULATION
- CONTINUOUS SIMULATION

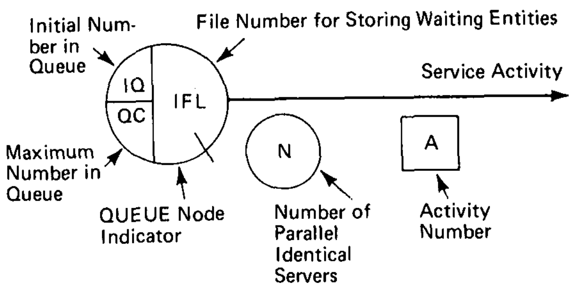
This slide attempts to illustrate the relative position of SLAM to other familiar simulation languages. Simulation languages are essentially either continuous or discrete. Those which are discrete are either process oriented such as GPSS, or event oriented such as SIMSCRIPT. Some of the familiar continuous models include CSMP and DYNAMO. SLAM essentially combines the elements of QGERT along with those of the GASP IV language. GASP IV was previously developed to combine continuous and event simulation. This marriage leads to a very powerful simulation language which I would recommend for your consideration.

WHY SLAM SIMULATION LANGUAGE?



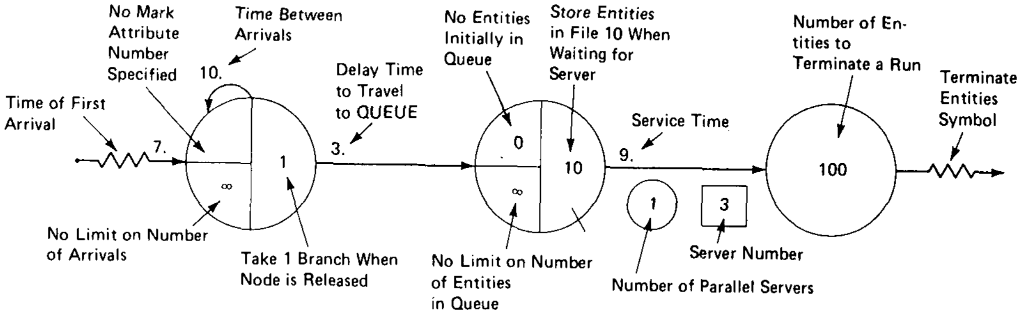
To demonstrate some of the features of the SLAM language, this network representation can be considered. The basic element is the queue node which includes information dealing with the initial number in the queue, maximum number, priority for storing elements in the queue, number of parallel servers, and the like.





(from ref. 3)

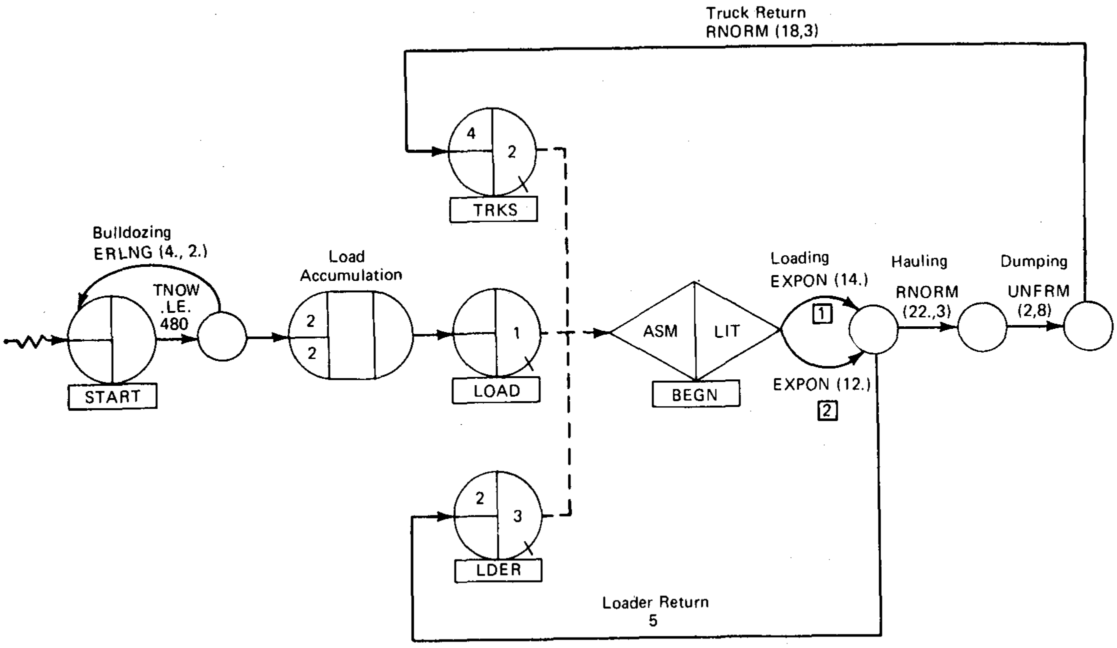
This slide is a model demonstrating the information available on the simplest of queueing models. This model involves the generation of elements and shows the capabilities of delaying arrivals, generating time between arrivals, and limiting the number of arrivals. From the queueing standpoint, it gives information with respect to the initial number in the queue, limit on the number in the queue and service time. The termination node is also shown to collect information on the number of units to be processed through the system. At the bottom of the slide, the language necessary to process this simple queueing system in SLAM is shown.



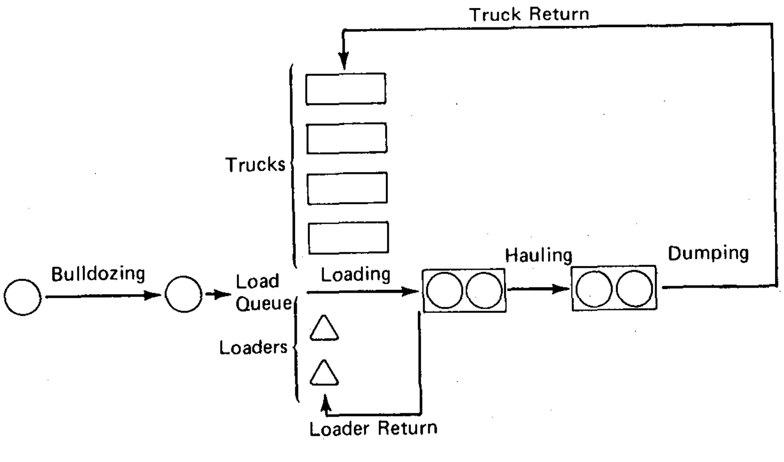
;EXAMPLE OF A SLAM STATEMENT MODEL	
NETWORK;	START OF NETWORK STATEMENTS
CREATE,10.,7.;	TIME BETWEEN ARRIVALS = 10
ACTIVITY,3;	TIME TO REACH QUEUE NODE IS 3
QUEUE(10);	USE FILE 10 FOR QUEUE
ACTIVITY(1)/3,9;	SERVICE TIME = 9
TERMINATE,100;	RUN MODEL FOR 100 ENTITIES
ENDNETWORK;	END OF NETWORK STATEMENTS

(from ref. 3)

The next slide presents a somewhat more interesting and complicated model and demonstrates some of the flexibility available in SLAM network modeling. The system essentially involves a truck hauling problem. Bulldozers collect material and two loads of dirt are necessary to fill a truck. Before a truck can be filled, there must be a loader, a truck and appropriate material to be loaded. The ASM node in this model is an assembly node and it says there must be at least one time in the truck queue, load queue, and the loader queue before the truck can be loaded. The load accumulation node requires that two loads must be available before a load moves into the load queue. The LIT represents the selection of servers which are the service time to load the product. After the product has been loaded, the loader returns to the loader queue and the truck moves to the dumping area. Finally, the truck returns to the trucking queue. This model could be used to determine the appropriate number of trucks, loaders and bulldozers necessary to accomplish an effective program. While this is not a space project, I believe analyzing it demonstrates some of SLAM's potential for analyzing NASA's progress.



SLAM network model of truck hauling system.



A truck hauling situation.

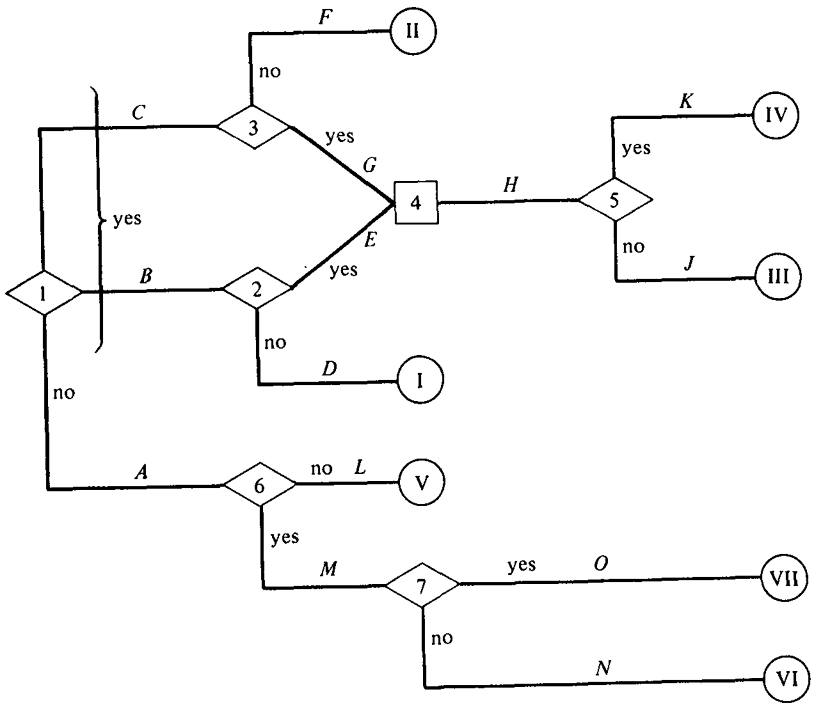
(from ref. 3)

There have been many SLAM applications to date, some of which include work station analysis, machine tool breakdowns, traffic problems, quarry operations, job shop scheduling, hospital modeling, and pilot ejection, which involve the network, continuous, and discrete analysis.

TYPICAL SLAM APPLICATIONS

- WORK STATIONS IN SERIES
- MACHINE TOOL BREAKDOWNS
- TRAFFIC LIGHT MODEL
- QUARRY OPERATIONS
- JOB SHOP SCHEDULING
- PSYCHIATRIC WARD
- PILOT EJECTION
- WORLD DYNAMICS
- SOAKING PIT FURNACE

In the spirit of the decision modeling theme of this conference, two final topics were selected to consider. The network shown on this slide is a model proposed a number of years ago by Graham to analyze research and development expenditures using a network type approach. The definitions of the events and outcomes are given on this slide. For each branch, Graham gives the probability that the branch is realized given that the preceding node has been realized, the time and the cost associated with the activity.



R & D expenditures.

Events:

1. Feasibility study indicates whether electrical control of high temperature system is feasible.
2. Determination of the suitability of ac control.
3. Determination of the suitability of dc control.
4. Optimum integration of ac/dc units achieved.
5. and 7. Unit economic feasibility of the design.
6. Determination of feasibility of pneumatic control.

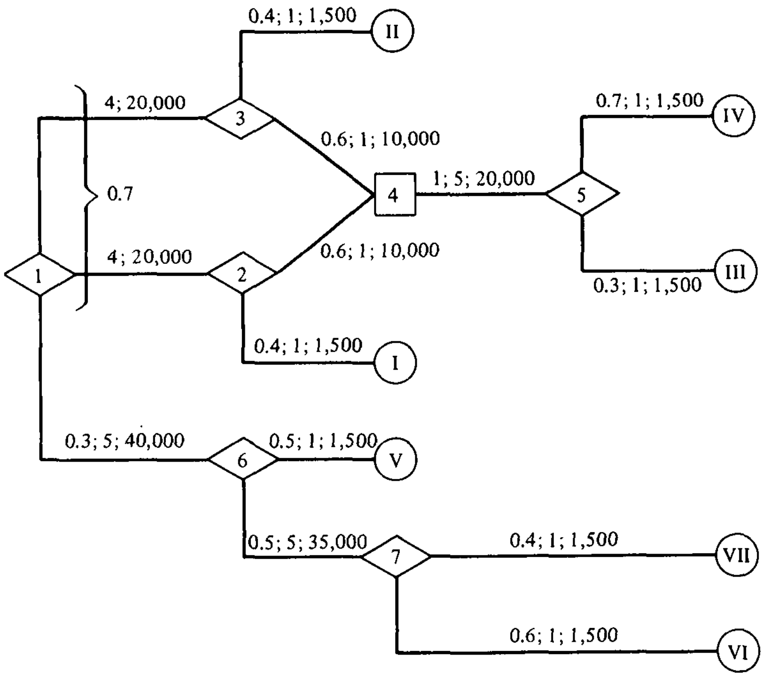
Activities:

(*A*) Pneumatic feasibility study.
(*B*) ac control investigation.
(*C*) dc control investigation.
(*D, F, J, K, L, N,* and *O*) Report writing.
(*E* and *G*) Investigation of optimum ac/dc integration.
(*H* and *M*) Economic analysis.

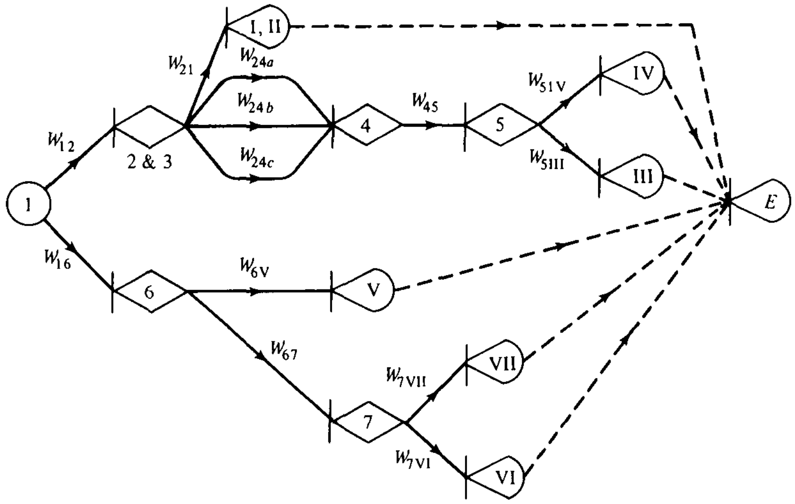
Outcomes:

(I, II, III, V, VI) Project dropped.
(IV, VII) Project into production and marketed.

The network at the top of the next slide shows information that Graham assumed for his research and development project. These values are inserted in the GERT network shown at the bottom of the page. Several changes were made in changing to GERT format. First, the AC/DC control investigations are performed simultaneously and thus are indicated on the network without the aid of a bracket. Second, nodes I and II do not result in the project being dropped as implied by the previous figure. The analysis of this network has interesting results.



R & D project with costs and time shown.



where

$W_{12} = 0.7e^{4s+40m}$,			$W_{5IV} = 0.7e^{s+1.5m}$,
$W_{16} = 0.3e^{5s+40m}$,			$W_{5III} = 0.3e^{s+1.5m}$,
$W_{21} = 0.16e^{s+3m}$,			$W_{6V} = 0.5e^{s+1.5m}$,
$W_{24a} = 0.36e^{s+20m}$,			$W_{67} = 0.5e^{5s+35m}$,
$W_{24b} = 0.24e^{s+11.5m}$,			$W_{7VI} = 0.6e^{s+1.5m}$,
$W_{24c} = 0.24e^{s+11.5m}$,		and	$W_{7VII} = 0.4e^{s+1.5m}$.
$W_{45} = e^{5s+20m}$,			

GERT representation of a R & D project.

Analysis of the GERT network associated with Graham's analysis allowed us to determine the expected costs associated with a project along with the expected time of completion. The probabilities of marketing electronic and pneumatic devices are also able to be calculated.

$$\begin{aligned} M_{1,E}(s,m) = {} & 0.7e^{4s+40m}(0.16e^{s+3m}(0.36e^{s+20m} + 0.48e^{s+11.5m})(e^{5s+20m}) \\ & \times (0.7e^{s+1.5m} + 0.3e^{s+1.5m})) \\ & + 0.3e^{5s+40m}(0.5e^{s+1.5m} + 0.5e^{5s+35m}) \\ & \times (0.4e^{s+1.5m} + 0.6e^{s+1.5m})) \end{aligned}$$

The expected cost of the project will equal $\partial M_{1,E}(s,m)/(\partial m)\big|_{m=0}^{s=0}$.

$$\left.\frac{\partial M_{1,E}(s,m)}{\partial m}\right|_{\substack{s=0\\m=0}} = 67.582$$

This is interpreted as 67.582 thousands of dollars, which checks with the published work of Graham.

The expected time of the project will equal $\partial M_{1,E}(s,m)/(\partial s)\big|_{m=0}^{s=0}$.

$$\left.\frac{\partial M_{1,E}(s,m)}{\partial s}\right|_{\substack{s=0\\m=0}} = 9.58 \text{ months}$$

This quantity is the expected duration of the project, which was not calculated by Graham.

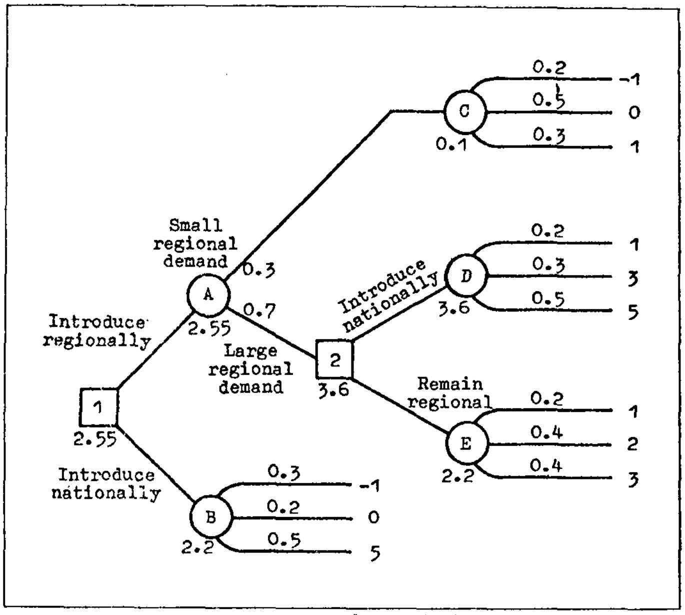
The probability of marketing an electronic control device is $W_{1,IV}(s,m)\big|_{m=0}^{s=0}$.

$$W_{1,IV}(s,m)\big|_{m=0}^{s=0} = (0.7)(0.84)(1.0)(0.7) = 0.4116$$

The probability of marketing a pneumatic device is equal to $W_{1,VII}(s,m)\big|_{m=0}^{s=0}$.

$$W_{1,VII}(s,m)\big|_{m=0}^{s=0} = (0.3)(0.5)(0.4) = 0.0600$$

The basis of a number of sequential decision making systems is the decision tree shown on this slide. The two basic elements are the decision node and a chance node. This particular decision tree involves the decision whether an item should be marketed nationally or regionally. While it is possible to effectively analyze a model of this sort, there has been some concern about information that might be overlooked. A number of years ago, I developed a technique called INFOCISION which attempted to incorporate some of the modeling ideas discussed in this presentation into the decision tree approach.



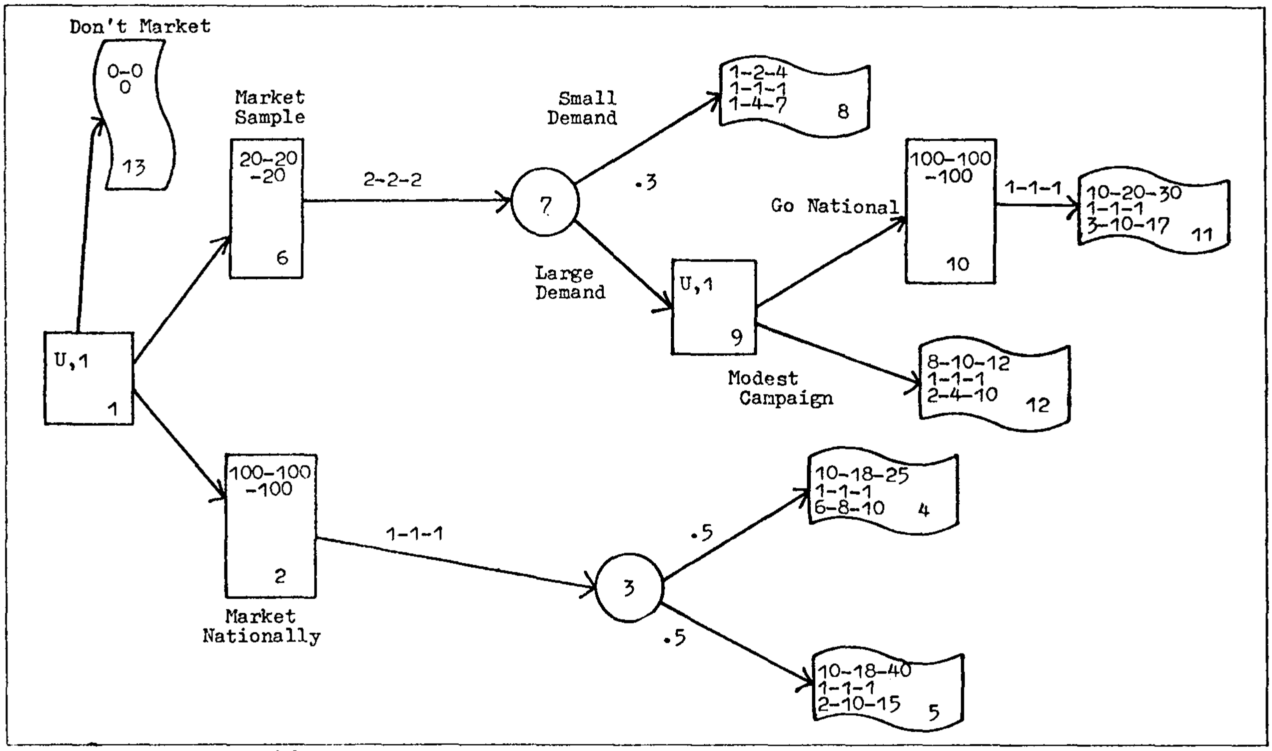
Decision tree analysis of a marketing model. The square nodes are "decision nodes," the round ones are "chance nodes."

The next slide shows some of the nodes I choose to use for the analysis INFOCISION model. Nodes include the decision node but allow for various decision criteria. The revenue and expense node allows for probabilistic data in much the same way PERT networks do. This system also allows for voting strategies.

TYPE	SYMBOL	PARAMETERS	DESCRIPTION
Decision (D)	C, P Node #	C = Criterion	Selects "Best" Alternative E = Max Expected Return U = Max Expected Utility A = Aspiration Level P = Most Probable Future
		P = Utility Curve #	Reference # for Criterion U only
		P = Aspiration Level	$ Amount Aspired to — for Criterion A only.
Chance (C)	Node #	None	Selects one alternative according to probabilities of activities emanating from this node.
Revenue (R)	$-A,M,B R-A,M,B T-A,M,B Node #	$ = First year Revenue Estimates	Optimistic, Most Likely, and Pessimistic Estimates of Revenue during first year.
		R = Annual Change Rate Estimates	Optimistic, Most Likely, and Pessimistic Estimates of Annual Growth (or Decline) Rate.
		T = Duration Estimates	Optimistic, Most Likely, and Pessimistic Estimates of number of years that revenue will last.
Revenue (R)	$-A,M,B Node #	$ = Revenue Estimates	Optimistic, Most Likely, and Pessimistic Estimates of one-time revenue.
Expense (E)	$-A,M,B R-A,M,B T-A,M,B Node #	$ = First year Expense Estimates	Analogous to Revenue Parameters
		R = Annual Change Rate Estimates	
		T = Duration Estimates	
Expense (E)	$-A,M,B Node #	$ = Expense Estimates	Analogous to Revenue Parameters
Vote (V)	P_1,P_2 Node #	P_1= Condition for First Realization	The number of inflowing activities which must occur before this node is realized for the first time (and emanating activities released).
		P_2= Condition for Subsequent Realizations	The number of additional inflowing activities which must occur before this node is realized (and emanating activities released) for all times subsequent to the first.
Construction (N)	Node #		Nonfunctional. For building purposes, and statistic collection purposes only.

Node description used in the Info-Cision technique.

This slide demonstrates the modification of the network previously analyzed for a marketing selection and shows the differences that might exist. This model allows for discounting of money and also allows for different types of decision criteria. In this particular case, a utility decision criterion will be used and information regarding the time for marketing sampling was built into the model. While it is not particularly important to consider the results of this model, it is interesting to note that this technique has been applied to a number of government and industrial applications. It may be that there are some opportunities to incorporate this philosophy into some of NASA's future decision modeling exercises.



Info-Cision model of the marketing problem. Dollars are expressed in thousands and the rate of return is 12 percent.

This slide lists five important references which I believe would be useful in your further work in the stochastic network area. I would be personally happy to interact with any NASA individual who might be interested in pursuing the topics discussed here and to try to relate them to NASA's problems. I want to thank you very much for inviting me to present my thoughts to you, and I hope that I have fostered some interest on your part in the stochastic network area.

REFERENCES

1. WHITEHOUSE - Systems Analysis and Design Using Network Techniques Prentice-Hall 1973.
2. PRITSKER - Modeling and Analysis Using Q-GERT Networks Halsted Press 1977.
3. PRITSKER and PEGDEN - Introduction to Simulation and SLAM Halsted Press 1979.
4. ELMAGHRABY - Activity Networks Project Planning and Control by Network Models Wiley-Interscience 1977.
5. WHITEHOUSE- "Using Decision Flow Networks" Industrial Engineering July 1974.

RECENT RESEARCH IN NETWORK PROBLEMS WITH APPLICATIONS

GERALD L. THOMPSON
CARNEGIE-MELLON UNIVERSITY

The decade of the 1970's witnessed a stunning improvement in the computability and applicability of network programming problems. Current applications involve problems having thousands or millions of variables and thousands of constraints. Network codes developed in the 1970's are 100 - 200 times as fast as their predecessors.

The main purpose of this talk is to survey the capabilities of network codes and their extensions to specially structured integer programming problems which can be solved by using the solutions of a series of ordinary network problems.

Most of the actual computational methods and results surveyed in this talk are taken from papers or working papers by the author and his students. This was done because of their easy availability. Many other authors have contributed important ideas to this area which we do not have time to discuss here. References to their work are given in the bibliography. It is not the purpose of this talk to give a historically accurate and complete account of the advances in network modeling and computing, so that the bibliography will have to suffice as a substitute for such a historical account.

FACTORIES (SOURCES) $I = \{1,\ldots,M\}$

SUPPLIES A_I FOR $I \in I$

MARKETS (SINKS) $J = \{1,\ldots,N\}$

DEMANDS B_J FOR $J \in T$

ASSUME

$$\sum_{I \in I} A_I = \sum_{J \in J} B_J$$

SEMIASSIGNMENT PROBLEM

$B_J = 1$ FOR $J \in J$

ASSIGNMENT PROBLEM

$A_I = 1$ FOR $I \in I$, AND

$B_J = 1$ FOR $J \in J$

SLIDE 1

SLIDE 1 gives the basic notation for a transportation problem having m factories and n markets. The factories have supplies a_i and the markets have demands b_j. Note that the sum of the supplies is assumed to be equal to the sum of the demands. When the demands are 1's the problem is called a <u>semi-assignment problem</u>; and when the supplies are also 1's, it is called an <u>assignment problem</u>.

SLIDE 2 gives the transportation problem constraints which the variables x_{ij} must satisfy. The first constraint says that the total amount shipped from warehouse i is equal to the amount it contains; the second constraint says that the sum of the amounts shipped to market j is equal to its demand; the last constraint is just non-negativity. The (bipartite) graph at the bottom shows that the direction of shipping is from factories to markets.

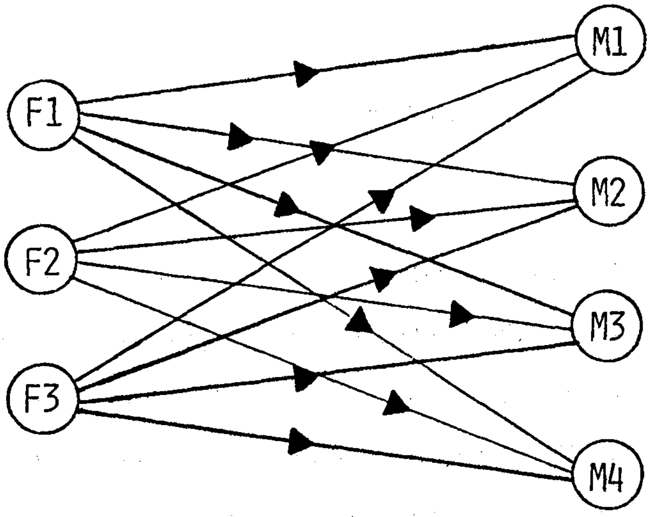
x_{IJ} = AMOUNT SHIPPED FROM FACTORY I TO MARKET J.

CONSTRAINTS

$$\sum_{J \in \mathrm{J}} x_{IJ} = A_I \qquad \text{FOR } I \in \mathrm{I}$$

$$\sum_{I \in \mathrm{I}} x_{IJ} = B_J \qquad \text{FOR } J \in \mathrm{J}$$

$$x_{IJ} \geq 0 \qquad \text{FOR } I \in \mathrm{I},\ J \in \mathrm{J}.$$



SLIDE 2

Slide 3 gives the two kinds of objective functions we will consider. The sum objective adds together all the shipping costs from each warehouse to each market. It is appropriate for bulk shipments of non-perishable goods. The bottleneck objective is appropriate when c_{ij} is interpreted as the time to ship goods from factory i to market j, and the objective is to minimize the maximum time along any route which carries a positive shipment. The bottleneck objective is appropriate when considering problems such as shipping perishable goods to markets in which we are concerned with the longest time for any shipment to get to its destination, or sending troops to staging areas in a case in which the unit is not ready to go until all sub-units have achieved their starting positions.

OBJECTIVE FUNCTIONS

<u>SUM OBJECTIVE</u>

c_{IJ} = COST OF SHIPPING ONE UNIT FROM I TO J.

MINIMIZE $\left\{ Z = \sum_{I \in I} \sum_{J \in J} x_{IJ} c_{IJ} \right\}$

Z = TOTAL SHIPPING COST

EXAMPLE: GROCERY WAREHOUSES TO SUPERMARKETS

<u>BOTTLENECK OBJECTIVE</u>

c_{IJ} = TIME TO SHIP ONE UNIT FROM I TO J

MINIMIZE $\left\{ Z = \underset{x_{IJ} > 0}{\text{MAXIMUM}} \; c_{IJ} \right\}$

Z = MAXIMUM SHIPPING TIME

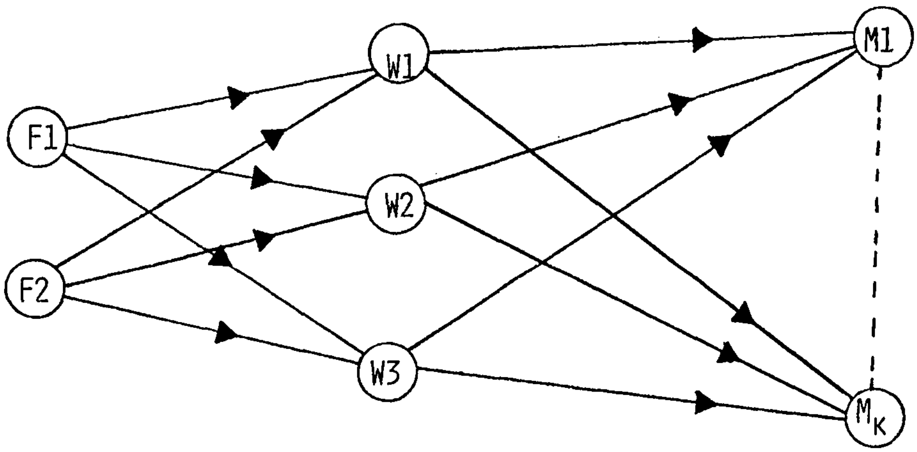
EXAMPLES: PERISHABLE GOODS, STAGING OF TROOPS

SLIDE 3

Slide 4 introduces the concept of a transshipment node; that is, one which is both a source and a sink. Network problems are transportation problems in which most nodes are transshipment nodes. Usually, not all of the possible arcs connecting pairs of nodes are assumed to exist in network problems; that is, the problems are sparse. Very large sparse problems have been formulated and solved relatively quickly.

<u>TRANSSHIPMENT NODE</u>: ONE THAT APPEARS BOTH AS A SOURCE AND AS A SINK.

<u>EXAMPLE</u>: FACTORY-WAREHOUSE-MARKET SYSTEM



NETWORK PROBLEMS: MOST NODES ARE TRANSSHIPMENT.

SPARSE: NOT ALL ARCS ARE USED. CAN BE TAKEN ADVANTAGE OF.

SLIDE 4

FAST PRIMAL METHODS FOR SOLVING BOTH SUM AND BOTTLENECK PROBLEMS WERE DEVELOPED IN THE 1970's.

TYPICAL RESULTS: 100 x 100 DENSE PROBLEMS.

	COST CHOSEN IN INTERVAL			
OBJECTIVE	0-10	0-100	0-1000	0-10000
SUM	.438	2.090	2.351	2.419
BOTTLENECK	.490	1.229	1.123	1.023

NOTE THAT BOTTLENECK PROBLEMS ARE APPROXIMATELY ONE HALF AS DIFFICULT AS SUM PROBLEMS.

NOTE MINIMUM COST EFFECT.
SPARSE PROBLEMS CAN BE SOLVED MUCH FASTER.
1000 x 1000 SPARSE PROBLEMS CAN BE SOLVED IN LESS THAN 2 MINUTES.

MUCH BIGGER PROBLEMS HAVE BEEN SOLVED
50,000 x 50,000

SLIDE 5

The idea of computational complexity is in vogue among computer science and OR practitioners. As noted on Slide 6, transportation and network problems are among the easiest such problems since they are polynomially bounded; that is, in the worst case the maximum number of steps required to solve such a problem can be constrained by a bound which is a polynomial function of the amount of input data needed. Transportation problems are natural integer problems since they will have integer solutions when the a_i's and b_j's are integers. For both these reasons, these problems are important in applications.

THERE ARE SEVERAL POLYNOMIALLY BOUNDED PRIMAL ALGORITHMS FOR THE SUM PROBLEM.

FORD-FULKERSON DUAL METHOD

BALINSKI-GOMORY PRIMAL METHOD

SRINIVASAN-THOMPSON COST OPERATOR METHOD

THE SRINIVASAN-THOMPSON-SZWARZ-HAMMER ALGORITHM CAN BE SHOWN TO BE POLYNOMIALLY BOUNDED.

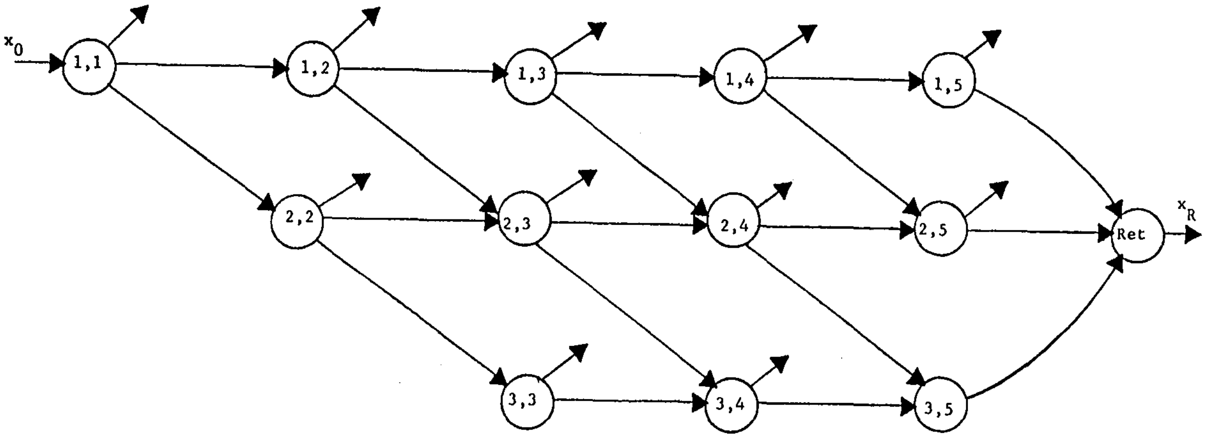
THOMPSON HAS A NEW RECURSIVE METHOD FOR BOTH SUM AND BOTTLENECK PROBLEMS WHICH IS POLYNOMIALLY BOUNDED.

ALSO, IF THE A_I's AND B_J's ARE INTEGRAL THEN A BASIC PRIMAL FEASIBLE INTEGER SOLUTION WILL BE FOUND BY THESE PRIMAL ALGORITHMS. THIS IS A <u>NATURAL INTEGER PROBLEM</u>.

THE NATURAL INTEGER PROPERTY MAKES THESE MODELS USEFUL FOR APPLICATIONS.

SLIDE 6

Slide 7 shows a typical network application in the area of manpower planning. Here there are three ranks and a maximum of five years of organizational age. Separations from the organization are indicated by upward slanting arrows, promotions by downward slanting arrows, and continuations in rank by horizontal arrows. The full model also has upper bounds on flows in each of these arcs. Note that there is one source node and one sink node (retirement), and all other nodes are retirement nodes. This is fairly typical for a network application.



State version of the manpower model with $R = 3$ and $T = 5$. Upward slanting arrows denote separations, horizontal arrows denote continuation in rank, and downward slanting arrows denote promotions. The yearly number of new employees is x_0, the yearly number of retirements is x_R, and the yearly separations (sum of flows on all upward slanting arrows) is x_S; we require $x_R + x_S = x_0$.

SLIDE 7

Slide 8 shows a fairly typical warehouse (or factory) to market application which is a straightforward transportation problem application. Unfortunately, many such applications also have other constraints which are not transportation type constraints. The single source constraint at the bottom of the slide is one such. It imposes the very commonly occurring requirement that all the demand at a given market be supplied from a single warehouse. We discuss methods for imposing such constraints next.

EXAMPLE 2. WAREHOUSES TO MARKETS

$$
\begin{array}{c|cccc|c}
 & M1 & M2 & \cdots & M_N & \\
\hline
W1 & c_{11} & c_{12} & \cdots & c_{1N} & A_1 \\
W2 & c_{21} & c_{22} & \cdots & c_{2N} & A_2 \\
. & & & \cdots & & . \\
W_M & c_{M1} & c_{M2} & \cdots & c_{MN} & A_M \\
\hline
 & B_1 & B_2 & \cdots & B_N &
\end{array}
$$

IF WE ADD OTHER CONSTRAINTS WE USUALLY DESTROY THE NATURAL INTEGER PROPERTY.

EXAMPLE: SINGLE (SOLE) SOURCE CONSTRAINT

$$X_{IJ} = \begin{cases} 0 \\ B_J \end{cases}$$

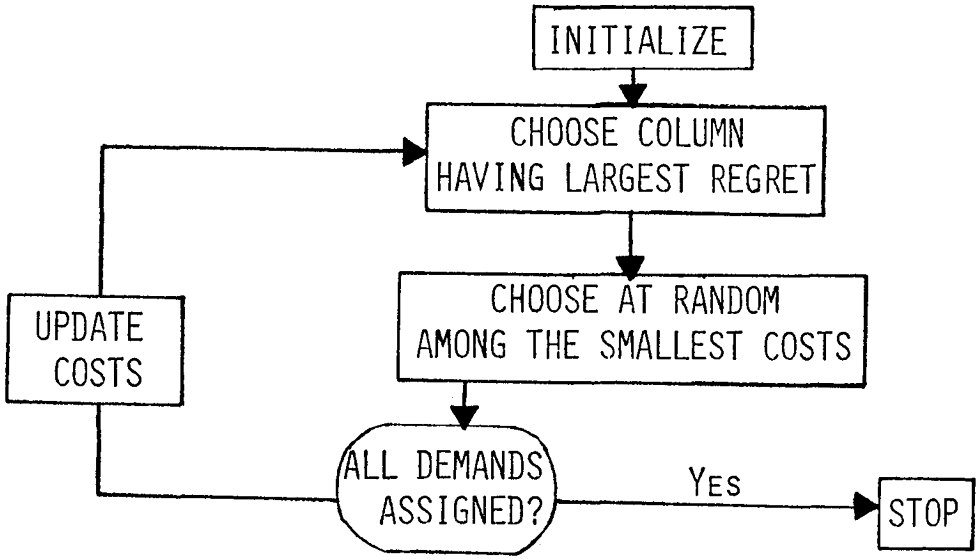
I.E., ALL THE DEMAND AT A MARKET MUST BE SUPPLIED FROM A SINGLE WAREHOUSE.

SLIDE 8

Slide 9 gives a computational flow diagram of the Regret Heuristic which (sometimes) finds good feasible solutions to single source problems. Note that it contains a random choice element so that each time it is run a (potentially) different solution is found. This heuristic is not guaranteed to get a feasible solution, but it usually does, and about half of the time the feasible solution is also optimal.

SINGLE SOURCE REGRET HEURISTIC

$\text{REGRET}_J = (\text{Second smallest entry} - \text{Smallest entry})_J$



RUN THIS PROGRAM SEVERAL (SAY 10) TIMES: SAVE BEST SOLUTION FOUND.

SLIDE 9

In contrast to a *heuristic code* which can only produce feasible solutions, an *algorithm* is a code which will, if it is run long enough, produce an optimal solutión to a decision problem (when such an optimal solution exists). Slides 10 and 11 discuss some of the concepts needed to implement a branch and bound algorithm for solving the single source problem. On Slide 10, note that the first thing we do is to relax the integer single source constraints to be just nonnegativity constraints. The relaxed problem is an ordinary transportation problem whose solution value gives a lower bound on the value of the unrelaxed problem. Usually, the relaxed solution will not satisfy all the single source constraints; variables which violate these constraints are called *fractional variables.* We choose one such variable and *branch*; that is, we consider the two subproblems in which the fractional variable is set either to zero or to the total demand of the column it is in. We relax the remaining variables in these two subproblems and solve them as transportation problems to get their lower bounds. This process is continued until we get either a feasible solution which allows us to update the upper bound (UB), or else we obtain a lower bound greater than an already achieved upper bound and can terminate search on this branch of the search tree--the latter step is also called *fathoming.* A typical search tree is shown in Slide 11.

BRANCH AND BOUND ALGORITHM

RELAX THE SINGLE SOURCE CONSTRAINT

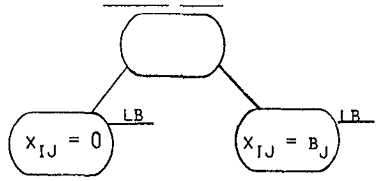
$$x_{IJ} = \begin{cases} B_J \\ 0 \end{cases} \quad \text{TO} \quad x_{IJ} \geq 0$$

SOLVE THE RESULTING TRANSPORTATION PROBLEM. ITS VALUE GIVES A LOWER BOUND ON THE VALUE OF THE SINGLE SOURCE PROBLEM

FIND A COLUMN WITH A FRACTIONAL VARIABLE, I.E.,

$$0 < x_{IJ} < B_J$$

BRANCH: DEVELOP THE SEARCH TREE



SOLVE EACH TRANSPORTATION PROBLEM TO GET LOWER BOUNDS (LB).

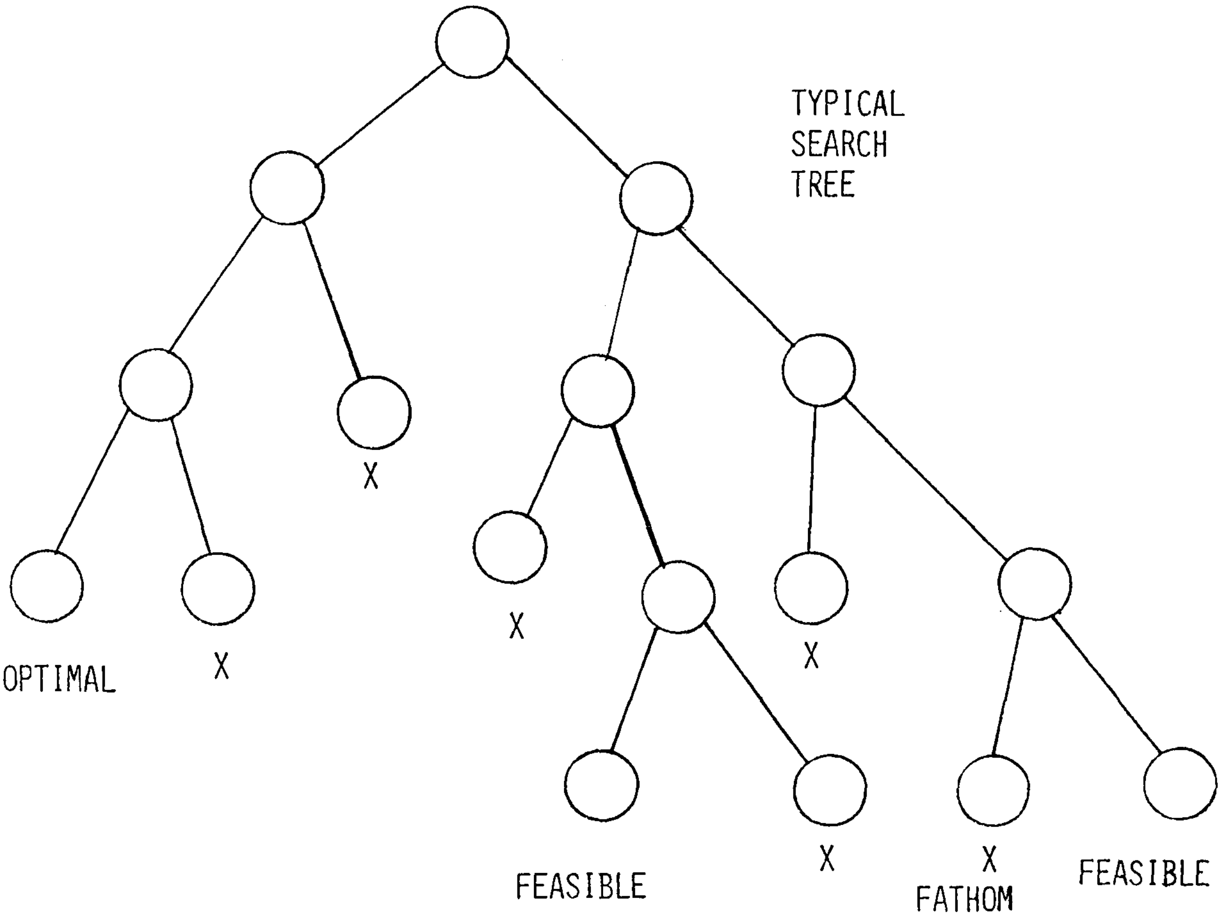
SLIDE 10

BRANCH AND BOUND (CONT)

WHENEVER LB $\geq$ UB FATHOM, I.E., DON'T SEARCH LOWER IN THE TREE.

WHENEVER A FEASIBLE SOLUTION IS FOUND (SATISFYING SINGLE SOURCE CONDITIONS) WITH A BETTER BOUND UPDATE UB

WHEN SEARCH IS COMPLETE HAVE OPTIMAL SOLUTION

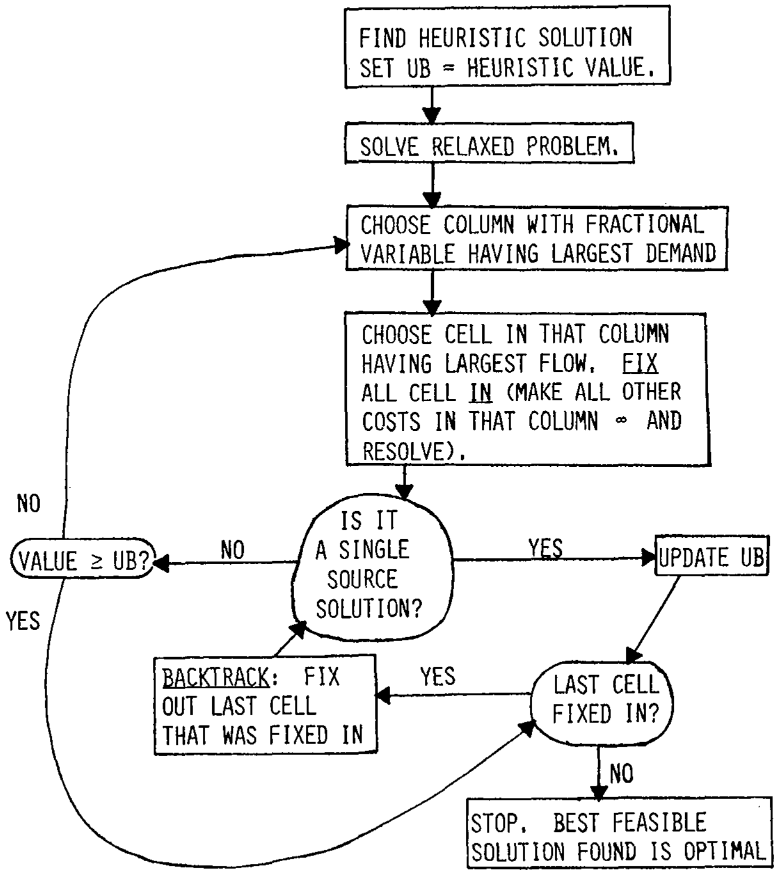


SLIDE 11

Slide 12 gives a LIFO (Last In First Out) or depth first branch and bound algorithm for solving the single source transportation problem. Note that it begins by finding a heuristic solution to give the initial value of UB, the upper bound. Then the relaxed problem is solved. Then a column having a fractional variable is selected and one of the cells having largest flow is selected; it is <u>fixed in</u>, that is, made to supply the total demand. If the resulting solution is single source, we update UB and <u>backtrack</u>, that is, go upward on the search tree. If it is not a single source solution, we test to see if the value of the subproblem is ≥ UB to see if we can fathom. If we can fathom, we backtrack--otherwise we choose another fractional variable and search deeper in the tree. The computational procedure stops when we try to backtrack from the initial node of the search tree.

LIFO BRANCH AND BOUND ALGORITHM

FOR THE SINGLE SOURCE PROBLEM



SLIDE 12

As noted at the top of Slide 13, the forward and backtrack movements in the search tree are actually performed by using cost operators which are computationally inexpensive. Also shown there are computation times obtained recently by Nagelhout and Thompson. Note that the heuristic frequently finds the optimum. Also note that solution times vary erratically depending on the size of the search tree. In one case, computation was stopped because of excessive time. These are typical results for this kind of problem.

MOVING UP AND DOWN THE SEARCH TREE
IS DONE BY APPLYING COST OPERATORS,
A TYPE OF PARAMETRIC PROGRAMMING.

COMPUTATIONAL RESULTS FOR SINGLE-SOURCE SUM PROBLEMS

M X N	HEURISTIC		NUMBER OF SEARCH TREES NODES	TOTAL TIME (SECS)
	% ERROR	CPU TIME		
100 x 100	0	2.11	505	23.88
100 x 200	3.3	4.38	946	48.07
100 x 300	0	7.71	20	18.75
100 x 350	-	9.94	6749	7600
100 x 400	0	10.78	3	23.39

COMPUTATIONAL TIMES FOR SINGLE-SOURCE BOTTLENECK PROBLEMS

M X N	HEURISTIC GET OPTIMAL?	NUMBER OF SEARCH TREE NODES	TOTAL TIME (SECS)
100 x 100	No	46	19.1
100 x 150	No	9263	115.4
100 x 400	Yes	-	12.48
100 x 400	No	133	106

SLIDE 13

Slides 14 and 15 discuss the Travelling Salesman problem which can be solved by similar procedures. The rubber band heuristic for the travelling salesman proceeds as follows. Choose any three cities and find their smallest subtour; now choose any city omitted and try inserting it in between pairs of cities on the subtour so far constructed; continue until a complete tour is obtained. The relaxed problem is an assignment problem which, if solved, will usually have loops or subtours. To develop a branch and bound code we solve the relaxed (assignment) problem, select a smallest subtour, choose an arc on that subtour and fix it out; now solve the new relaxed problem and iterate until a feasible tour is found, then backtrack, etc. The rest of the code is similar to that for the single source problem.

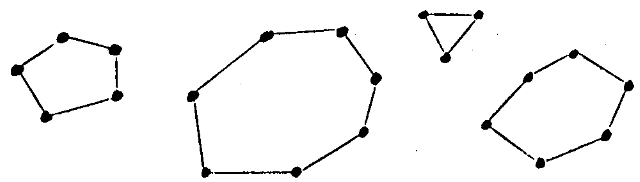
TRAVELLING SALESMAN PROBLEM

GIVEN N CITIES FIND A ROUTE THAT GOES THROUGH EACH CITY EXACTLY ONCE AND MINIMIZES THE TOTAL MILEAGE TRAVELLED (OR, MINIMIZES THE MAXIMUM INTERCITY DISTANCE.)

HEURISTIC SOLUTION: RUBBER BAND HEURISTIC.

PROBLEM RELAXATIONS: ASSIGNMENT PROBLEM OR BOTTLENECK ASSIGNMENT PROBLEM

RELAXED PROBLEM HAS SUBTOURS.



BRANCHING RULE. CHOOSE A SMALLEST SUBTOUR AND BRANCH ON SOME ARC IN IT.

SLIDE 14

Slide 15 gives some computation times for sum and bottleneck travelling salesman problems. Note that for the sum case, the total solution time goes up rapidly with the number of cities, but the average time to the first tour (which is usually within a few percent of the optimum) remains small. The first tour found by the algorithm can be used as an improved heuristic solution. The most amazing results are for the bottleneck travelling salesman problem; Smith and Thompson have solved such problems for up to 2,000 cities. The reason that this is possible is that the search trees remain surprisingly small as noted at the bottom of Slide 15.

RANDOMLY GENERATED ASYMMETRIC SUM PROBLEMS COSTS (0-100)
(SMITH-SRINIVASAN-THOMPSON, 1977)

NO. CITIES	50	100	150	180
AVE. TIME TO OPTIMALITY	1.72	52.98	65.28	617.12
AVE. TIME TO FIRST TOUR	.6	5.2	9.0	23.00

TIMES ARE MEASURED ON UNIVAC-1108.

FIRST TOURS ARE ALWAYS WITHIN 5% OF OPTIMUM
AND USUALLY MUCH CLOSER

BIVALENT (COSTS 0-1) PROBLEMS WITH 200 CITIES SOLVED
IN LESS THAN 6 SECS.

RANDOMLY GENERATED BOTTLENECK PROBLEMS
(SMITH-THOMPSON, 1975)

NO. CITIES	200	500	1000	1500	2000
AVE. TIME TO OPTIMALITY	2.75	20.08	33.72	206.43	343.87
AVE. NO. OF NODES IN SEARCH TREE	6.6	16.8	7	16	12

SLIDE 15

The last major example to be discussed is the capacitated warehouse location problem stated in Slide 16. Note that the x_{ij} variables are as before, but the y_i variables take on only the integer values 1 if warehouse i is open, and 0 if it is closed. In the objective function a fixed charge F_i is added when warehouse i is opened. The relaxed problem here is obtained by just requiring $y_i \geq 0$, i.e., nonnegativity. We do not discuss further details of the branch and bound code.

CAPACITATED WAREHOUSE LOCATION

VOTER REDISTRICTING

PROBLEM

$$\text{MINIMIZE} \left\{ \sum_{I \in I} \sum_{J \in J} X_{IJ} C_{IJ} + \sum_{I \in I} F_I Y_I \right\}$$

$$\sum_{J \in J} X_{IJ} = A_I Y_I$$

$$\sum_{I \in I} X_{IJ} = B_J$$

$$X_{IJ} \geq 0 \qquad Y_I = \begin{cases} 0 & \text{WAREHOUSE I CLOSED} \\ 1 & \text{WAREHOUSE I OPENED} \end{cases}$$

PROBLEM RELAXATION

TRANSPORTATION PROBLEM

	1	N	N+1	
1	C_{11}	C_{1N}	0	A_I
M	C_{1M}	C_{MN}	0	A_M
	B_1	B_N	B_{N+1}	

SLIDE 16

Slide 17 gives computational results obtained by Nagelhout and Thompson on this problem. Note that the bottleneck problems are much easier than the sum problems, since the bottleneck code has no failures while the sum code failed to solve two sum objective problems. It is also true that the variance of times is much less for the bottleneck than for the sum objective problems.

The problems discussed so far are far from exhausting the applications of network and transportation problems.

WAREHOUSE LOCATION PROBLEMS

SUM OBJECTIVE

SIZE	SEARCH TREE NODES	TIME (SECS)
15 x 50	35	3
25 x 50	500	30 (2 FAILURES)
15 x 45	1000	35

BOTTLENECK OBJECTIVE

SIZE	SEARCH TREE NODES	TIME (SECS)
15 x 50	102	2.4
30 x 90	214	5.3
50 x 150	457	17.0

NO FAILURES

SLIDE 17

Slide 18 lists 8 other application areas which will be briefly discussed. Also discussed are two other network models. The first is a network with gains in which the quantity of the good can increase or decrease as it flows along an arc. An example of an increase is: Suppose the commodity is money and flowing on the arc means being on deposit in a savings account for a period of time; the money can then be augmented by an interest payment. An example of a decrease is: Suppose the quantity is electrical power flowing in a wire; it can be decreased due to power losses. The final generalization is to multi-commodity flows in which we consider several commodities flowing on the same arc and competing for its capacity.

OTHER APPLICATIONS

1. κ-TOUR TRAVELLING SALESMAN
2. OPTIMAL GROWTH PATHS
3. CASH MANAGEMENT MODEL
4. ELECTRICAL POWER DISTRIBUTION
5. ELECTRICAL POWER CAPACITY PLANNING
6. TRANSPORTATION WITH STOCHASTIC DEMANDS
7. DECISION CPM
8. CLUSTER ANALYSIS

EXTENSIONS

1. NETWORKS WITH GAINS
2. MULTI-COMMODITY FLOW MODEL

SLIDE 18

References

1. Abraham, J., "Uber die Stabilitat der Losungen im Transportproblem der Linearen Programmierung," Czechosl. Math. Journal, Vol. 8 (83), pp. 131-138, 1958.

2. Balachandran, V., "Optimal Production Growth for the Machine Loading Problem," Naval Research Logistics Quarterly, Vol. 22, No. 3, pp. 593-607, September 1975.

3. ____________, "An Integer Generalized Transportation Model for Optimal Job Assignment in Computer Networks," Operations Research, Vol. 24, No. 4, pp. 742-759, July-August 1976.

4. ____________, "The Stochastic Generalized Transportation Problem--An Operator Theoretic Approach," Management Sciences Research Report No. 311, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, February 1973.

5. ________________ and S. Jain, "Optimal Facility Location Under Random Demand with General Cost Structure," Naval Research Logistics Quarterly, Vol. 23, No. 3, pp. 421-436, September 1976.

6. ________________ and A. Perry, "Transportation Type Problems with Quantity Discounts," Naval Research Logistics Quarterly, Vol. 23, No. 2, pp. 195-209, June 1976.

7. _________________ and G.L. Thompson, "An Operator Theory of Parametric Programming for the Generalized Transportation Problem - I: Basic Theory," Naval Research Logistics Quarterly, Vol. 22, No. 1, pp. 79-100, March 1975.

8. ____________ and ____________, "An Operator Theory of Parametric Programming for the Generalized Transportation Problem - II: Rim, Cost and Bound Operators," Naval Research Logistics Quarterly, Vol. 22, No. 1, pp. 101-125, March 1975.

9. __________ and __________, "An Operator Theory of Parametric Programming for the Generalized Transportation Problem - III: Weight Operators," Naval Research Logistics Quarterly, Vol. 22, No. 2, pp. 297-315, June 1975.

10. __________ and __________, "An Operator Theory of Parametric Programming for the Generalized Transportation Problem - IV: Global Operators," Naval Research Logistics Quarterly, Vol. 22, No. 2, pp. 317-339, June 1975.

11. Balas, E. and P. Ivanescu (P.L. Hammer), "On the Transportation Problem - I and II," Cahiers du Centre de Recherche Operationnelle (Brussels), Vol. 4, Nos. 2 and 3, pp. 98-116 and 131-160, 1962.

12. Bowman, E.H., "Production Scheduling by the Transportation Method of Linear Programming," Operations Research, Vol. 4, No. 1, pp. 100-103, February 1956.

13. Bradley, G.H., G.G. Brown and G.W. Graves, "Design and Implementation of Large-Scale Primal Transshipment Algorithms," Management Science, Vol. 24, No. 1, pp. 1 - 34, September 1977.

14. Caccetta, L., "The Transportation Problem with Piecewise Linear Costs," Working Paper, Department of Mathematics, University of Western Australia, 1976.

15. Charnes, A. and W.W. Cooper, Management Models and Industrial Applications of Linear Programming, John Wiley and Sons, Inc., New York, Vol. 1, 1961.

16. __________ and M. Kirby, "The Dual Method and the Method of Balas and Ivanescu for the Transportation Model," Cahiers du Centre de Recherche Operationnelle (Brussels), Vol. 6, No. 1, pp. 5-18, 1964.

17. Dantzig, G.B., Linear Programming and Extensions, Princeton University Press, Princeton, N.J., 1963.

18. __________ and P. Wolfe, "Decomposition Principle for Linear Programs," Operations Research, Vol. 8, No. 1, pp. 101-111, January-February 1960.

19. Eastman, W.L., "Linear Programming with Pattern Constraints," Unpublished doctoral dissertation, Harvard University, Boston, Massachusetts, 1958.

20. Eisemann, K., "The Generalized Stepping Stone Method for the Machine Loading Model," Management Science, Vol. 11, No. 1, pp. 154-176, September 1964.

21. Fong, C.O. and V. Srinivasan, "Multiperiod Capacity Expansion and Shipment Planning with Linear Costs," Naval Research Logistics Quarterly, Vol. 23, No. 1, pp. 37-52, March 1976.

22. _________ and _________, "Determining all Nondegenerate Shadow Prices for the Transportation Problem," Transportation Science, Vol. 11, No. 3, pp. - , August 1977.

23. _________ and _________, "The Multi-Region Dynamic Capacity Expansion Problem - I and II," Research Reports #227-228, Graduate School of Business, Stanford University, Stanford, California, 1974.

24. Ford, L.R. and D.R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, New Jersey, 1963.

25. Garfinkel, R.S. and M.R. Rao, "The Bottleneck Transportation Problem," Naval Research Logistics Quarterly, Vol. 18, No. 4, pp. 465-472, December 1971.

26. Garstka, S.J., "Computation in Stochastic Programs with Recourse," unpublished doctoral thesis, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1970.

27. Gass, S.I. and T. Saaty, "The Computational Algorithm for the Parametric Objective Function," Naval Research Logistics Quarterly, Vol. 2, No. 1, pp. 39-45, March 1955.

28. _________ and _________, "Parametric Objective Function - Part II: Generalization," Operations Research, Vol. 3, No. 4, pp. 395-401, November 1955.

29. Gelders, L. and P.R. Kleindorfer, "Coordinating Aggregate and Detailed Scheduling Decisions in the One-Machine Job Shop: Part I - Theory," Operations Research, Vol. 22, No. 1, pp. 46-60, January-February 1974.

30. Gensch, D.H. and E.A. Kervinen, "The Traveling Salesman's Subtour Problem," Working Paper, School of Business Administration, University of Wisconsin, Milwaukee, Wisconsin, 1975.

31. Glover, F., D. Karney, D. Klingman and A. Napier, "A Computational Study on Start Procedures, Basis Change Criteria, and Solution Algorithm for Transportation Problems," Management Science, Vol. 20, No. 5, pp. 793-813, January 1974.

32. ___________ and D. Klingman, "New Advances in the Solution of Large Scale Network and Network Related Problems," Research Report #CS-177, Center for Cybernetic Studies, The University of Texas, Austin, Texas, 1974.

33. Grabowski, W. and W. Szwarc, "Optimum Transportation Bases," Zastosowania Matematyki (Warsaw), Vol. 9, No. 4, pp. 357-389, 1968.

34. Hammer, P.L., "Time-Minimizing Transportation Problems," Naval Research Logistics Quarterly, Vol. 16, No. 3, pp. 345-357, September 1969.

35. Held, M. and R. M. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees: Part II," Mathematical Programming, Vol. 1, No. 1, pp. 6-25, February 1971.

36. Holt, C., F. Modigliani, J. Muth and H. Simon, Planning Production, Inventories and Workforce, Prentice-Hall, Englewood Cliffs, New Jersey, 1960.

37. Mulvey, J.M., "Network Relaxations for 0-1 Integer Programs," Paper No. WA 3.4 presented at the Joint National Meeting of the Operations Research Society of America and The Institute of Management Sciences, Las Vegas, Nevada, November 1975.

38. Perry, A., "An Optimum Seeking Algorithm for Locating Lock Boxes," Midwest AIDS Proceedings, April 1975.

39. Rao, R.C. and D.P. Rutenberg, "Multilocation Plant Sizing and Timing," Management Science, Vol. 23, No. 11, pp. 1187-1198, July 1977.

40. Shapiro, D.M., "Algorithms for the Solution of the Optimal Cost and Bottleneck Traveling Salesman Problems," unpublished doctoral thesis, Washington University, St. Louis, Missouri, 1966.

41. Simmonard, M., Linear Programming, Prentice-Hall, Englewood Cliffs, N.J., 1966.

42. Smith, T.H.C., V. Srinivasan and G.L. Thompson, "Computational Performance of Three Subtour Elimination Algorithms for Solving Asymmetric Traveling Salesman Problems," Annals of Discrete Mathematics, Vol. 1, pp. 495-506, 1977.

43. Srinivasan, V., "A Transshipment Model for Cash Management Decisions," Management Science, Vol. 20, No. 10, pp. 1350-1363, June 1974.

44. ____________ and G.L. Thompson, "An Operator Theory of Parametric Programming for the Transportation Problem - I," Naval Research Logistics Quarterly, Vol. 19, No. 2, pp. 205-225, June 1972.

45. ________ and ________, "An Operator Theory of Parametric Programming for the Transportation Problem - II," Naval Research Logistics Quarterly, Vol. 19, No. 2, pp. 227-252, June 1972.

46. ________ and ________, "Determining Optimal Growth Paths in Logistics Operations," Naval Research Logistics Quarterly, Vol. 19, No.4, pp. 575-599, December 1972.

47. ________ and ________, "An Algorithm for Assigning Uses to Sources in a Special Class of Transportation Problems," Operations Research, Vol. 21, No. 1, pp. 284-295, January-February 1973.

48. ________ and ________, "Benefit-Cost Analysis of Coding Techniques for the Primal Transportation Algorithm," Journal of the Association for Computing Machinery, Vol. 20, No. 2, pp. 194-213, April 1973.

49. ________ and ________, "Solving Scheduling Problems by Applying Cost Operators to Assignment Models," in S.E. Elmaghraby (Ed.) Symposium on the Theory of Scheduling and its Applications, Springer-Verlag, NY, pp. 399-425, 1973.

50. __________ and __________, "Algorithms for Minimizing Total Cost, Bottleneck Time, and Bottleneck Shipment in Transportation Problems," Naval Research Logistics Quarterly, Vol. 23, No. 4, pp. 567-595, December 1976.

51. __________ and __________, "Determining Cost vs. Time Pareto-Optimal Frontiers in Multi-Modal Transportation Problems," Transportation Science, Vol. 11, No. 1, pp. 1-19, February 1977.

52. __________ and __________, "Cost Operator Algorithms for the Transportation Problem," Mathematical Programming, Vol. 12, No. 3, pp. 372-391, June 1977.

53. Szwarc, W., "The Stability of the Transportation Problem," Mathematica, Vol. 4 (27), No. 2, pp. 397-400, 1962.

54. __________, "The Time Transportation Problem," Zastosowania Matematyki, Vol. 8, No. 3, pp. 231-242, 1966.

55. Thompson, G.L., "Recent Theoretical and Computational Results for Transportation and Related Problems," in Modern Trends in Logistics Research, W.H. Marlow, Ed., MIT Press, 1976.

56. Veinott, A.F., "Lattice Programming," Working Paper (in process), Department of Operations Research, Stanford University, Stanford, California, 1977.

57. Wagner, H.M., Principles of Operations Research with Applications to Managerial Decisions, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.

58. Zeleney, M., Linear Multiobjective Programming, Springer-Verlag, New York, 1974.

59. Nagelhout, R. V. and G. L. Thompson, "A Single Source Transportation Algorithm,' Management Sciences Research Report No. 429, Carnegie-Mellon University, February 1979.

60. ______________ and ______________, "A Study of the Bottleneck Single Source Transportation Problem," Management Sciences Research Report No. 456, Carnegie-Mellon University, March, 1980.

61. ______________ and ____________, "A Cost Operator Approach to Multistage Location-Allocation," Management Sciences Research Report No. 440, Carnegie-Mellon University, July, 1979.

62. ______________ and ____________, "A Bottleneck Facility Location Problem," working paper, Carnegie-Mellon University, April, 1980.

SEQUENTIAL DECISION MAKING AND STOCHASTIC NETWORKS

SALAH ELMAGHRABY
NORTH CAROLINA STATE UNIVERSITY

Because of the interest expressed to me by various participants in the subject of scheduling, I shall preface my talk with some introductory remarks on this vibrant (and rather infant) field. My objective is to give a brief "state of the art" review.

Scheduling usually refers to the assignment of time intervals to the execution of tasks. This is typically accomplished by assigning start times and completion times to each task. Such assignment is subject to a variety of constraints (such as earliest ready times, precedence relations among tasks, non-splitting limitations, machine (or processor) capacity bounds, etc.), and is carried out to achieve a variety of objectives (relative to tardiness, flow time, number of required processors, setup and/or inventory costs, etc.).

At the dawn of the activity, optimization was the call of the day. Then, as the "easy" problems were exhausted - which occurred about the early sixties - it became obvious that another approach was needed, and heuristic scheduling became the fashion, especially in the so-called "job shop " scheduling problems in which no discernable pattern of job movement could be established. A typical endeavor would proceed as follows: the analyst first defines the problem, then suggests a heuristic, then "tests" it on a variety of problems of different parameters, and finally makes some statistical inferences about the "goodness" of his heuristic. I usually refer to these procedures as "unreliable heuristics" because their application guarantees no optimum, and the abortion of the procedure at any point of time yields no bounds on the error committed (= the difference between the best solution in hand and the (unknown) optimum).

These heuristic approaches stand in sharp contrast to implicit enumeration schemes (such as branch-and-bound procedures) that employ heuristics in several of their phases but, if permitted to run to termination, will yield the optimum.

I shall have more to say about heuristics in just a moment. Allow me first to finish this chapter of the story.

While confronting progressively more "difficult" problems, analysts noticed a strange phenomenon, namely, that the degree of "difficulty" seems to change rather abruptly. A problem that is easily optimized for two machines simply defies optimization for three machines under the very same criterion; and another problem that is easily resolved (optimally) under one criterion again defies solution under a miniscule (or seemingly miniscule) change in criterion!

The "tour de force" in this region came in 1971 when the theory of NP-completeness was born. This is neither the time nor the place to expound on this intriguing and controversial theory. Suffice it here to say that some scheduling problems (about 320 of them as of the moment, with additions almost daily) have been proven to be NP-complete, and that the solution of any one of them by an algorithm that runs in polynomial time (in the size of the input) would signal the solution of all scheduling problems in the class NP. Since several brilliant researchers have been trying their best to solve some of these problems for a number of years with no success (the literature on the

Traveling Salesman Problem should attest to this fact!), it seems highly improbable that these difficult problems will ever be solved by a "decent" algorithm.

This theory had two immediate impacts. First, it gave broad outline and a taxonomy to an area which, up to that time, did not have any. Second, it gave theoretical justification to the use of the above-mentioned, ignominious, "unreliable heuristics." For now, since there is little hope of achieving the optimum in "reasonable time," the field is left wide open to enterprising heuristicians!

SCHEDULING:

OPTIMIZATION

HEURISTICS

→ COMPLEXITY THEORY

FIGURE 1

I would like to mention three important points before leaving this subject.

First, the combinatorial theory of complexity (and NP-completeness) is a deep and important theory that helps us classify problems in the class NP. It also says something about the asymptotic behavior of problems: a problem that is known to be in the class NP-complete will require resources (time, computer capacity, etc.) that expand at a rate that is not bounded by any finite-order polynomial. But the theory gives us no clue to the solvability of small problems -this information must be gathered empirically, and depends on several local factors that include the type and size of the computer used, the expertise of the programmer, etc.

Second, the theory of combinatorial complexity spawned interest in heuristics from another point of view. I promised you before that I shall return to the subject of heuristics, and here I am! In particular, if a problem is known to be NP-complete, then perhaps a "good" heuristic (that runs in polynomial time) is the most we can hope for. But how to measure the "goodness" of a heuristic? Answer: by its worst case performance. Briefly, Heuristic A is preferred to Heuristic B if the <u>maximum</u> (relative) deviation from the optimum that results from the use of A <u>is smaller</u> than that of B. This criterion has two strikes against it: it is difficult to determine the "worst case performance" analytically, and, even if it were easily determinable, the criterion is too pessimistic. This latter objection is the age-old comment levied against the min-max loss criterion in classical decision theory. And it gave rise to a criterion that is based on the probabilistic behavior of the heuristic. Research in this area is still in its infancy and, therefore, I shall say no more about it.

Third, I certainly do not wish to leave the impression that the theory (of combinatorial complexity) is complete or that it answers all the questions asked. Far from it; the theory itself is in a feverish state of development, since the space of NP itself is not completely mapped out, let alone problems that are outside the space NP! On the other hand, everyday problems in scheduling have to be solved in some fashion, optimally or otherwise, and the theory of NP-completeness offers little help in this regard.

To sum up, scheduling at the practical level is forging ahead with heuristics and approximate procedures most of the time, and at the theoretical level with research in optimizing procedures for problems that have not yet been classified as NP-complete, as well as with research in completing the taxonomy of the field.

APPROXIMATION:

MEET PRE-SPECIFIED
ERROR

WORST CASE PERFORMANCE:

(HEURISTIC)

PROBABILISTIC BEHAVIOR

FIGURE 2

With almost half of my allocated time already gone, I would like now to turn to the original topic of my talk which is Sequential Decision Processes.

I would like to follow the outline exhibited in Figure 3, with special emphasis on needed research.

Sequential Decision Processes: Serial

- DP Formalism
- State Vector "Explosion"
- Multivariate Optimization
- Approximation: Orthog. Plyn.
  Splines

Non-Serial Systems

Infinite Horizon

Markov and Semi-Markov Processes

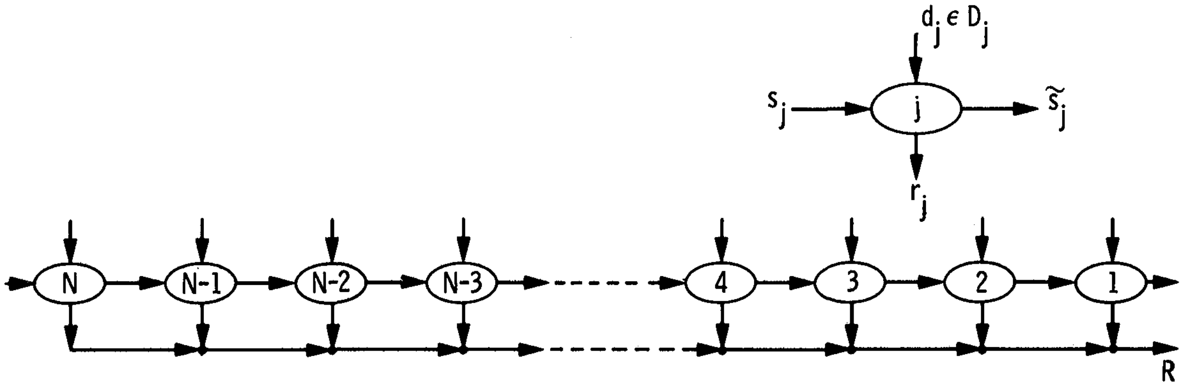
Figure 3

The basic mathematical tool of analysis is dynamic programming, which was originally developed for "serial" systems as shown in Figure 4, but which has since been developed to encompass diverging, converging, feedforward, and feedback branches, as shown in Figure 5.

The notation in these diagrams is as shown in the corner: each stage of the process has an input state s_j that is chosen from some decision space D_j, and a "reward" r_j, which may not be a reward at all but some excretion or product that bears no relation to the future states or decisions of the system, though it may be combined in some fashion with other "rewards" of previous or subsequent stages to yield the global "reward."

I would like to point out that the model depicted in these diagrams is quite general, and can easily accommodate many characteristics that are peculiar to any specific application. For one example, from a dimensional point of view, there need not be (and, in reality, it rarely happens that there is) any uniformity between the state, decision or reward. For another example, uncertainty is easily (from a conceptual point of view) accommodated in all three parameters: state, decision, and reward.

STRICTLY SERIAL PROCESS



$$f_j(s_j) = \underset{d_j \in D_j}{\text{Opt}} \left\{ r_j(s_j, d_j) \circ f_{j-1}(\tilde{s}_j) \right\}$$
$$\tilde{s}_j = t_j(s_j, d_j)$$

FIGURE 4

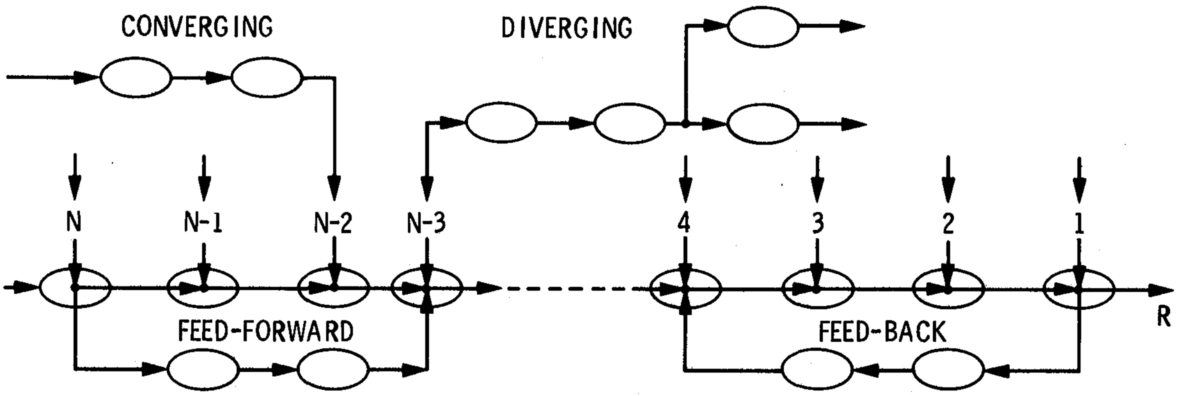


FIGURE 5

General as it may be, the model suffers from some important deficiencies that detract from its utility.

The first and most widely known is the so-called "curse of dimensionality" of the state vector; to wit, if the state is represented by a vector of dimensionality $n \geq 4$, then obtaining numerical answers becomes well-nigh impossible. There are two approaches that have been proposed as a means of resolution of this problem:

1. In discrete spaces, group (or "coagulate") subspaces into representative points and investigate only those points.

2. In continuous spaces, use orthogonal polynomials (which require, for their definition, an infinitesimal proportion of points at which the optimum is determined), or splines (up to cubic order).

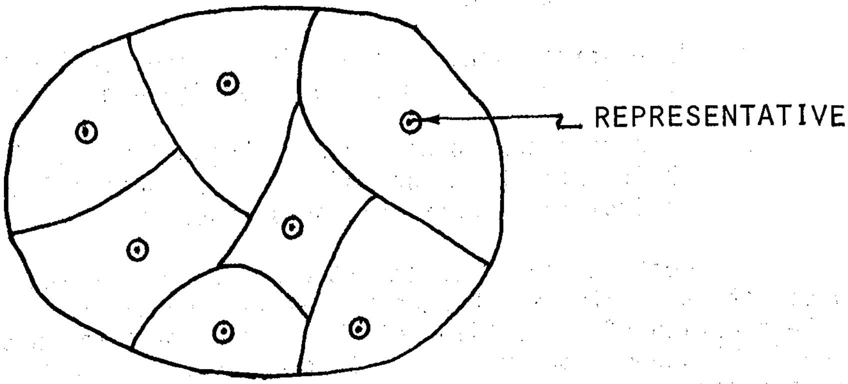
The most serious problem attached to both approaches is that of error propagation. This is inherent in the very nature of the DP approach: the determination of the optimum of the N-stage problem involves the optimum of the (N-1)-stage problem; hence, if the latter is only approximately known, so is the former. But, since the N-stage optimum is evaluated only at a few points and is approximate anywhere between these points, it is easy to see how the error in determining the value of the optimum at these points is compounded stage after stage.

PROBLEMS

1. CURSE OF DIMENSIONALITY OF STATE VECTOR

   A. DISCRETE SPACE:

      GROUP OR "COAGULATE" INTO SUBSETS



   B. CONTINUOUS SPACES:

      USE ORTHOGONAL POLYNOMIALS

      USE SPLINES

PROBLEM OF ERROR PROPAGATION

FIGURE 6

Another nasty problem facing the analyst is that of simultaneous optimization over several variables, as illustrated by $F_j(S_j)$ in Figure 7.

Now, this runs counter to the grain of DP which insists on "breaking up" the search over m-dimensional spaces into m searches each in one dimension only. Unfortunately, the device used to accomplish this end increases the dimensionality of the state vector, and we are back to the "curse of dimensionality" discussed before!

PROBLEMS (CONT'D)

2. SIMULTANEOUS OPTIMIZATION OVER SEVERAL VARIABLES

$$F_j(S_j) = \underset{D_{j1},\ldots,D_{jM}}{\text{OPT}} \left\{ R_j\,(S_j;D_{j1},\ldots,D_{jM})\ \text{O}\ F_{j-1}\,(\tilde{S}_j) \right\}$$

$$\tilde{S}_j = T_j(S_j;D_{j1},\ldots,D_{jM})$$

EXAMPLE OPT'L PROJECT "COMPRESSION"

$\{A_I\}$ - ACTIVITY I HAS: DURATION

COST

RELIABILITY

RESOURCE CONSUMPTION

G_I: $\text{DIM}_1\ (A_I) \longrightarrow \text{DIM}_2(A_I)$

DURATION $Y_I = G_I(X_I)$

INVESTMENT

PRECEDENCE

FIGURE 7

An excellent example of such eventualities can be found in the problem of "optimal project compression." I would like to spend a few minutes on this example, because it also ties our topic to another subject that is relevant to NASA activities; namely, Activity Networks.

Now, a project is a collection of activities that occur over time and consume resources, and are related by precedence. The duration y_i of activity i is a function of the amount of resources utilized by the activity, written as $y_i(x_i)$, where x_i is the "investment" in the resources. The function g_i may assume any shape, as shown in the top diagram of Figure 8, or it may be discontinuous as shown in the bottom diagram. If it assumes the latter shape, the following DP approach has been proposed for the optimal project "compression" under a specified budget.

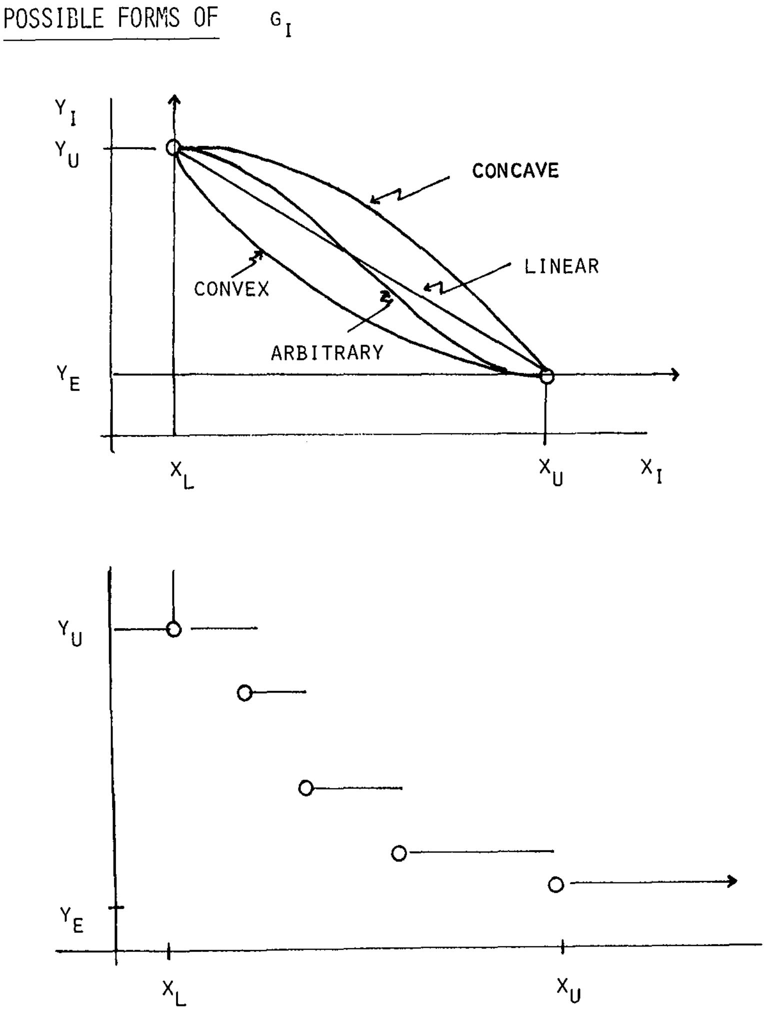
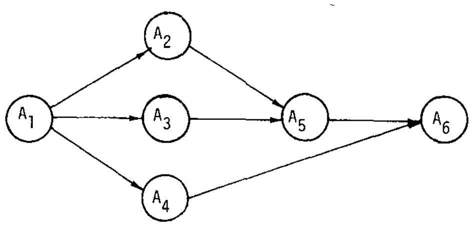


FIGURE 8

To render the approach clear, I shall illustrate it by reference to the network of Figure 9, in which there are six activities with the precedence shown. The functional equation of DP is also shown in Figure 9, and the DP iterative scheme follows in Figures 9 and 10. Everything works fine because the network is completely decomposable. The logic of the iterative scheme is depicted in Figure 11; but suppose it is not, such as in the miniscule network of Figure 12. A "straightforward" application of the DP formalism will lead to the simultaneous optimization over x_3 and x_5! Now, we must either perform this optimization, or else be obliged to "condition" on one of the decision variables; that is, add it to the state vector as exhibited in Figures 12 and 13. The logic of this process cannot be exhibited in an elegant (and simple) manner as was done in the previous Figure 11.

TOTAL AVAILABLE FUNDS = K



MINIMIZE TOTAL PROJECT DURATION ≡ MINIMIZE LONGEST PATH

$$\min_{X} \max \left\{ Y_1 + Y_2 + Y_5 + Y_6 \ ; \quad Y_1 + Y_3 + Y_5 + Y_6 \ ; \quad Y_1 + Y_4 + Y_6 \right\}$$

$$= \min_{X} \left\{ Y_1 + Y_6 + \max \left[Y_2 + Y_5 \ ; \ Y_3 + Y_5 \ ; \ Y_4 \right] \right\}$$

$$= \min_{X} \left\{ Y_1 + Y_6 + \max \left[Y_5 + \max \left(Y_2 \ ; \ Y_3 \right) \ ; \ Y_4 \right] \right\}$$

DP FORMALISM

$$F_1(k) = \min_{0 \le x_3 \le K} \quad G_3(x_3) \quad ; \quad 0 \le k \le K \le K$$

$$F_2(k) = \min_{0 \le x_2 \le K} \quad \max \left\{ G_2(x_2) \ ; \ F_1(k - x_2) \right\} ; \ 0 \le k \le K$$

FIGURE 9

$$F_3(K) = \min_{X_5} \left\{ G_5(X_5) + F_2(K-X_5) \right\};\ 0 \le K \le K$$

$$F_4(K) = \min_{X_4} \max \left\{ G_4(X_4)\ ;\ F_3(K - X_4) \right\};\ 0 \le K \le K$$

$$F_5(K) = \min_{X_6} \left\{ G_6(X_6) + F_4(K-X_6) \right\}\ ;\ 0 \le K \le K$$

$$F_6(K) = \min_{X_1} \left\{ G_1(X_1) + F_5(K-X_1) \right\}.$$

NOTE THAT NUMBER OF DP STAGES EQUALS NUMBER OF ACTIVITIES

⟵ COMPLETELY DECOMPOSABLE AN.

FIGURE 10

DP MODEL OF "COMPLETELY DECOMPOSABLE" ACTIVITY NETWORK

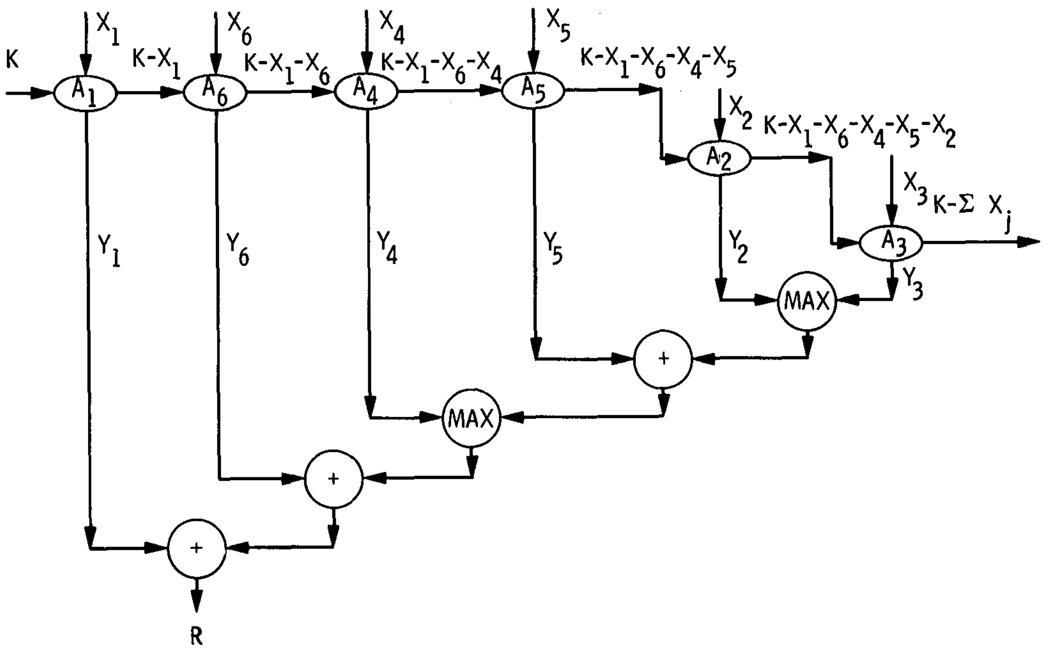
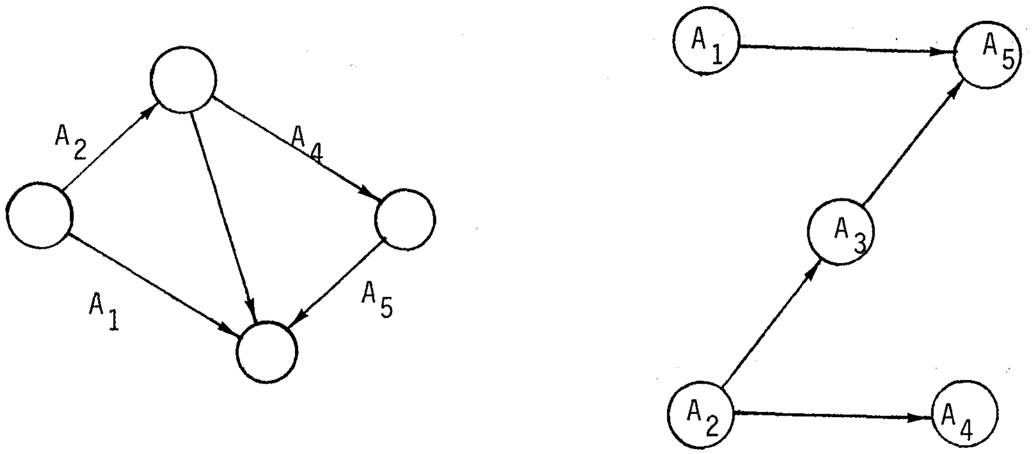


FIGURE 11



$$\min_{X} \max \left\{ Y_1 + Y_5 \ ; \ Y_2 + Y_3 + Y_5 ; \ Y_2 + Y_4 \right\}$$

$$= \min_{X} \max \left\{ Y_1 + Y_5 \ ; \ Y_2 + \max \left[Y_3 + Y_5 \ ; \ Y_4 \right] \right\}$$

TOTAL AVAILABLE FUNDS = K

FIX $\tilde{X}_5 \leq [0, K]$

$$F_1 (\kappa, \tilde{X}_5) = \min_{X_3} \left[G_3 (X_3) + \tilde{X}_3 \right] \qquad 0 \leq \kappa \leq K$$

$$F_2 (\kappa, \tilde{X}_5) = \min_{X_6} \max \left\{ G_4 (X_4), F_1 (\kappa - X_4 - \tilde{X}_5, \tilde{X}_3) \right\} \qquad 0 \leq \kappa \leq K$$

FIGURE 12

$$F_3\ (K,\tilde{X}_5) = \boxed{\min_{X_1,X_2}}\ \max\ G_1 \left\{ (X_1) + \tilde{Y}_5\ ;\ G_2\ (X_2) + \right.$$

$$\left. F_2\ (K - X_1 - X_2 - \bar{X}_5,\ \bar{X}_5) \right\}$$

$$\tilde{F}_3\ (K) = \min_{X_5}\ F_3\ (K,\ X_5)$$

ALTERNATIVE APPROACH: FIX $\tilde{X}_2$ IN STAGE 3

$$F_3\ (K,\ \tilde{X}_5,\ \tilde{X}_2) = \min_{X_1}\ \max \left\{ G_1\ (X_1) + \tilde{Y}_5\ ;\ \tilde{Y}_2 + \right.$$

$$\left. F_2\ (K - X_1 - \tilde{X}_2 - \tilde{X}_5,\ \tilde{X}_5) \right\}$$

$$\hat{F}_3\ (K,X_5) = \min_{X_2}\ F_3\ (K,\ \tilde{X}_5,\ \tilde{X}_2)$$

$$\hat{\hat{F}}_3\ (K) = \min_{X_5}\ \hat{F}_3\ (K,\ \tilde{X}_5)$$

FIGURE 13

So far, I have stated (and illustrated) a problem, but have not yet indicated any approach for its resolution. Since I am running out of time, let me just mention two avenues of attack that have been proposed:

1. Utilize concepts of dominance through bounding in the DP formalism to reduce the amount of computing required. Basically, this advocates the "marrying" of DP recursion and Branch-and-Bound methodology.

2. Relax the requirement of strict optimality in the search over the state space, and be content with a tolerable error.

Suffice it to say here that both approaches are still in their infancy; a great deal of work has to be done to place them on sound footings.

CONCEPTS OF DOMINANCE THROUGH BOUNDING:

MARRY DP & B & B

RELAXATION OF STRICT OPTIMALITY IN

SEARCH OVER STATE SPACE

FIGURE 14

DISTRIBUTED PROBLEM SOLVING AND NATURAL LANGUAGE UNDERSTANDING MODELS

CHARLES RIEGER
MARYLAND UNIVERSITY

I want to give a sampling of model building and systems related to natural, human understanding. I shall cover four topics of about 15 min. each. The topics are shown here going from the most theoretical to the most practical. Collectively they draw on the same ideas.

The needs of further development are given at the bottom.

WORD EXPERT PARSING

- SEMI-AUTOMATIC DIGITAL DESIGN
- CAUSAL MODELING AND MONITORING
- ZMOB

NEED:

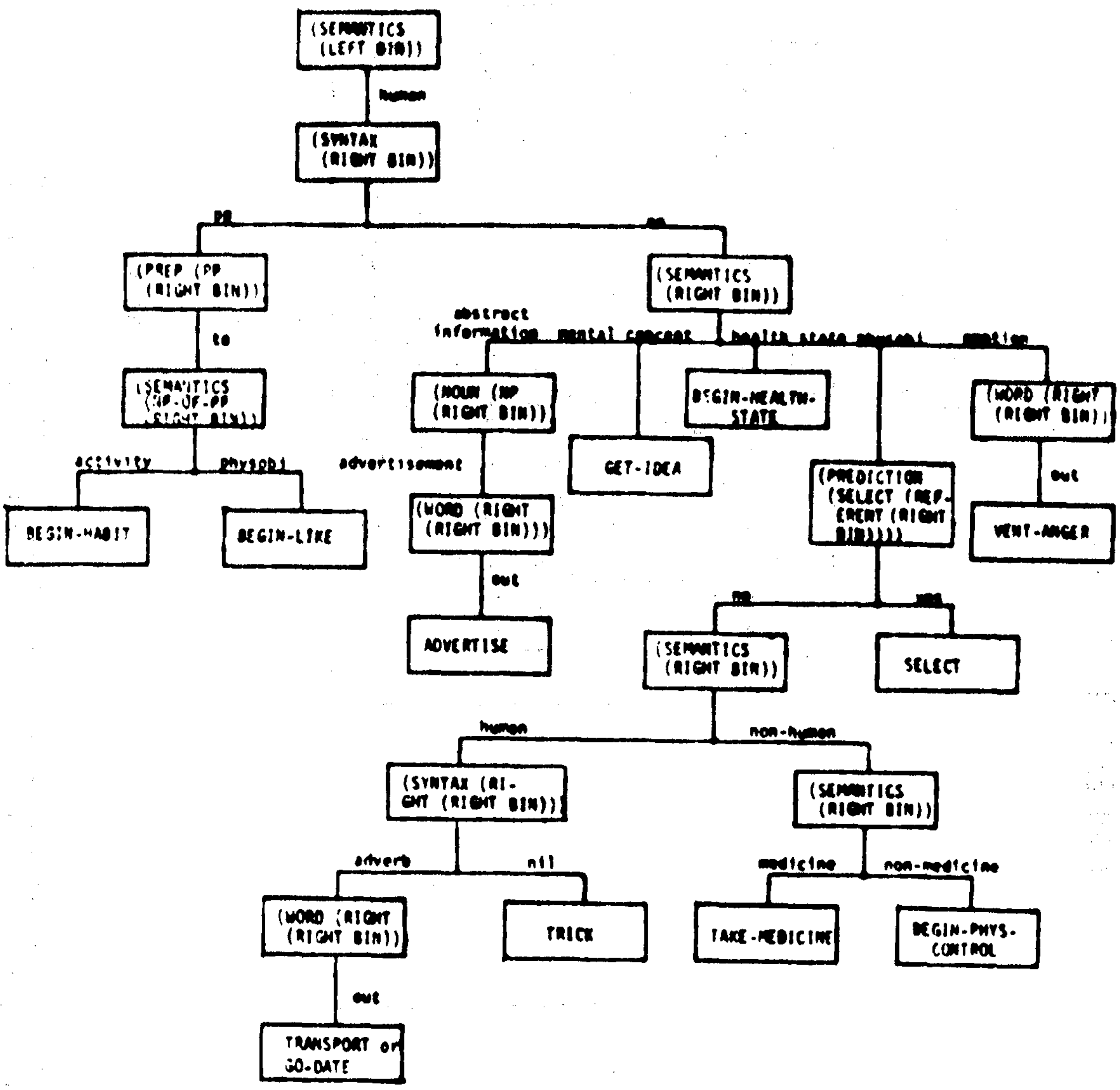
- MODELS OF NL UNDERSTANDING
- MODELS OF HUMAN UNDERSTANDING OF CAUSALITY
- MODELS OF GENERAL KNOWLEDGE STORAGE
- GOOD REAL-TIME DISPLAY-ORIENTED HUMAN INTERFACES

A few years ago, the most important issue to us was that of size of ambiguity and meaning of words. Somehow a fundamental notion seems to select the correct meaning of a word in a certain context. Every single word has usually many intepretations. For instance, we have here the verb "to take." To cope with this kind of complexity, we came up with the word "expert parser."

Some senses of the verb "to take"

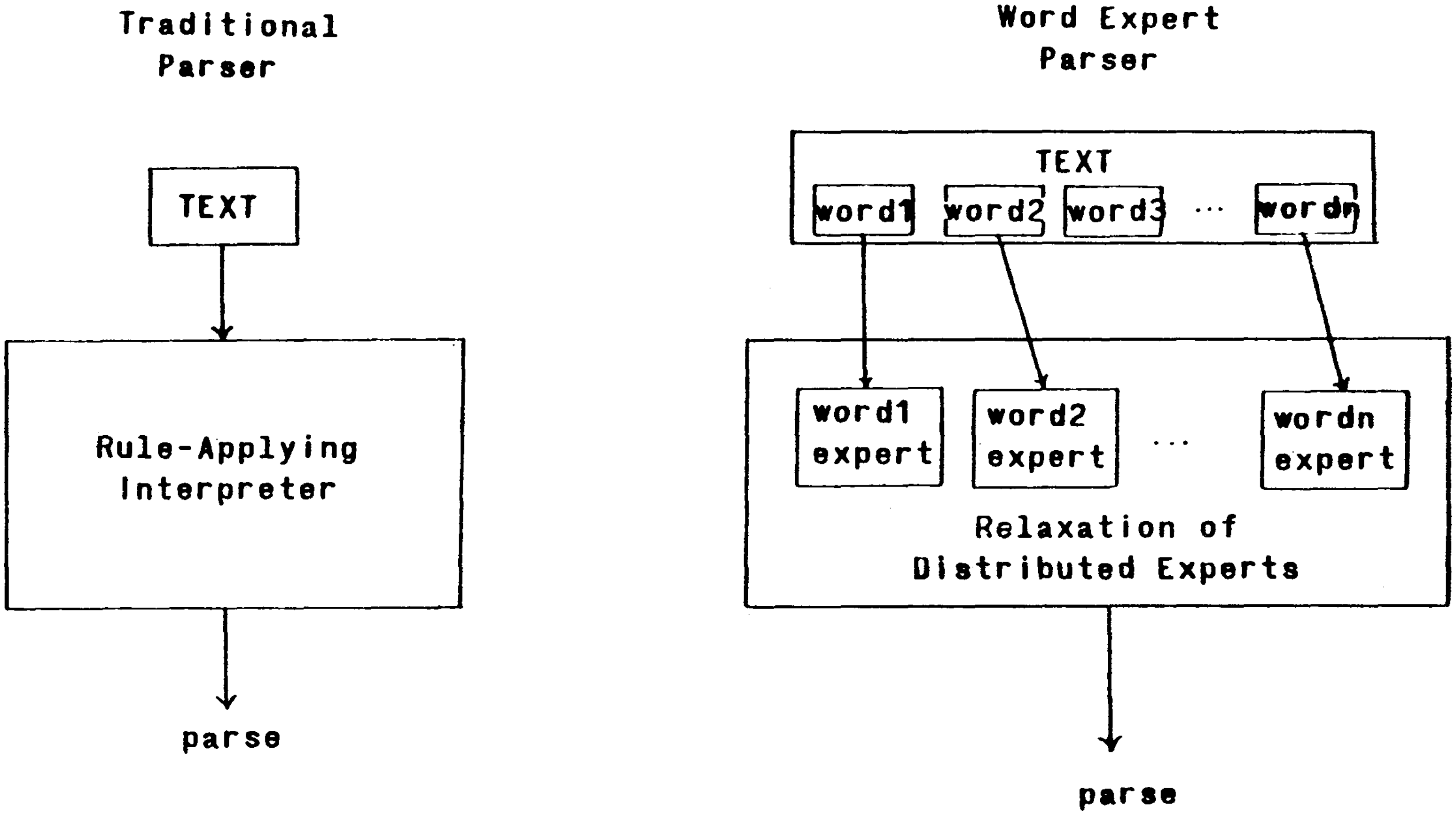
1 John took it easy.
2 John took a break.
3 John took a look.
4 John took a dive.
5 John took it like a man.
6 John took five.
7 John took a drive.
8 John took out the garbage.
9 John got the food to take out.
10 John took in a stranger.
11 John took her in his arms.
12 John took away the gun.
13 John took her for an idiot.
14 John took a letter.
15 John took to the road.
16 John took back the book.
17 John took her into his confidence.
18 John took leave of Mary.
19 John took a leave of absence.
20 John took two aspirins.
21 John took a flu shot.
22 John took her by the hand.
23 John took her hand.
24 John took everything but harmed no one.
25 John took notes in philosophy.
26 John took steno.
27 John took a dip.
28 The rent payment took John under.
29 That took a load off John.
30 John took a left.
31 John took off his clothes.
32 John took off for Mars.
33 John took off for the ball game.
34 John took the green one.
35 John only took an ounce.
36 John took a chance.
37 John took a leak.
38 John took grief from Mary.
39 John took over the department.
40 John took down the poster.
41 John took Mary for a ride.
42 John took the sucker for a ride.
43 John took down the message.
44 John took him to the mat.
45 John took back Mary.
46 John took a toke.
47 John took after his father.
48 John took upon the occasion to say a few words.
49 John took Mary around town.
50 John took his time.
51 John took care of Mary.
52 John took a nap.
53 John took a picture of Mary.
54 John took advice from Mary.
55 John took Mary as an example.
56 John took it out on Mary.

Here we have a discrimination tree or sense net for "take." This approach is very radical from what was being done before. Here the notion is: make every word by the language its own independent program. Instead of writing systems of rules, we are saying be more anarchistic about it but go the route of a highly distributed collection of experts. So we take the word "take" and think of all the possible meanings of the verb and put them into a construction called a word expert. If you have a sentence, you call up all your word experts from a disk which are LISP coroutines, then, on "go," they talk to each other independently, and form clusters, etc. until the sentence is parsed. As an example, we got 13 of us lined up in a row, gave everyone a 3x5 card with a word, and said "go." In about 20 minutes, the given sentence was parsed even though we had many ambiguous words in the sentence. The sentence was "The sudden death of the chimney sweeps threw the town in a state of shock." All our feelings about making every word an expert were enhanced. One can think of a lot of practical spin-offs.



ILLUSTRATIVE SENSE NET
FOR SOME SENSES OF THE VERB
"TAKE"

WORD EXPERT PARSING



Much of what has been said is on this slide. There are also other experts for punctuation symbols, which become rather important at times because they have semantic content.

This is an expert in LISP code for "pit." These are decision-encoded tree logics. We compile such things in CONNIVER style LISP code. We need fairly elaborate coroutine control.

```
(&expert pit
 (groups (PP sometimes n1)
         (ACT sometimes n2))
 (startnode n0)
 (network
  (n0 ((&expectp '(group PP)) n1)
      ((&concept (&leftc) 'PP) n5)
      (t (&susp-until-true '(&runp (&rightw))) n0))
  (n1 (t (&PP-construct) n2))
  (n2 ((&suspectp '(class LOCATION)) n3)
      ((&suspectp '(class VOLUME)) n3)
      ((&suspectp '(class FOOD)) n4)
      (t (&susp-until-true 't) n2))
  (n3 (t (&store (class volume))
         (&store (volume-type hole-in-ground))
         (&PP-close)
         (&terminate)))
  (n4 (t (&store (class FOOD))
         (&store (food-type FRUIT))
         (&store (fruit-type PART-OF))
         (&PP-close)
         (&terminate)))
  (n5 (t (&susp-until-true 'nil)))))

(&expert deep
 (groups (PA always n1))
 (startnode n0)
 (network
  (n0 (t (&PP-construct) n1))
  (n1 ((&suspectp '(class LOCATION)) n2)
      ((&terminatedp (&rightw)) n3)
      ((&runp (&rightw)) n4)
      (t (&susp-until-true '(&runp (&rightw))) n1))
  (n2 (t (&store (size LARGE))
         (&store (size-dimension DEPTH))
         (&constrain '(class VOLUME))
         (&terminate)))
  (n3 ((&descriptorp '(class HUMAN))
       (&store (intellectual-level HIGH))
       (&terminate))
      ((&descriptorp '(class VOLUME)) n2)
      (t (&susp-until-true 'nil)))
  (n4 (t (&constrain '(class VOLUME) '(class HUMAN))
         (&susp-until-true
          '(&terminatedp (&rightw))) n1))))

&expert in
(groups (T sometimes n1)
        (LOC sometimes n1))
(locals PP-db class)
(startnode n1)
(network
 (n1 (t (&T-LOC-construct)
        (&expect-rightc '(class SITUATION))
        (&susp-until-posted '(control-state UNKNOWN)) n2))
 (n2 (t (&T-LOC-resume)
        (cond ((eq class 'time) (&T-LOC-type class))
              (t (&T-LOC-type 'location)))
        (&T-LOC-close)
        (&terminate)))))
```

We do not have a monster system at this time. The system has about twenty words and each word has about five senses at this time. The most elaborate sentence we have parsed is "The deep philosopher throws the peach pit into the deep pit." It has a lot of interesting ambiguities in it. The following two slides show how the parsing is done.

```
.TQ EX%WORD4
'BPA ($$LAB)
     (PROG <$$NOTED *WD* *DB* GROUPS>
           <G$BINDALL>
           <G$GO1 $$LAB>
   $$START <SETQ *WD* 'WORD4>
           <SETQ *DB* NIL>
           <SETQ GROUPS '((PP SOMETIMES N1) (ACT SOMETIMES N2))>
        N0 <%TRACE-MSG "====> AT NODE: " 'N0>
           <COND [<&EXPECTP '(GROUP PP)> <GO N1>]
                 [<&CONCEPT (&LEFTC) 'PP> <GO N5>]
                 [T <GO $$LAB2>]>
    $$LAB2 <%QUEUE-EXPERT-SUSPEND (%RUNTIME (&RUNP (&RIGHTW))) 'WORD4
            >
           <UNFLAG 'WORD4 'SUSPENDED>
           <%REMPROP-BB 'WORD4 'STATUS '-OLD-STATUS>
           <FLAG 'WORD4 'SUSPENDED>
           <%PUT-BB 'WORD4 'STATUS 'SUSPENDED>
           <NOTE 'SUSPENDED>
           <%TRACE-MSG 'SUSPENDED " EXPERT: " 'WORD4>
           <AU-REVOIR $$LAB3>
    $$LAB3 <%REOPEN-DB *DB*>
           <GO N0>
        N1 <DO <%TRACE-MSG "====> AT NODE: " 'N1>
               <COND [T <&PP-CONSTRUCT>
                        <GO N2>]>>
        N2 <%TRACE-MSG "====> AT NODE: " 'N2>
           <COND [<&SUSPECTP '(CLASS LOCATION)> <GO N3>]
                 [<&SUSPECTP '(CLASS VOLUME)> <GO N3>]
                 [<&SUSPECTP '(CLASS FOOD)> <GO N4>]
                 [T <GO $$LAB5>]>
    $$LAB5 <%QUEUE-EXPERT-SUSPEND (%RUNTIME T) 'WORD4>
           <UNFLAG 'WORD4 'SUSPENDED>
           <%REMPROP-BB 'WORD4 'STATUS '-OLD-STATUS>
           <FLAG 'WORD4 'SUSPENDED>
           <%PUT-BB 'WORD4 'STATUS 'SUSPENDED>
           <NOTE 'SUSPENDED>
           <%TRACE-MSG 'SUSPENDED " EXPERT: " 'WORD4>
           <AU-REVOIR $$LAB6>
    $$LAB5 <%REOPEN-DB *DB*>
           <GO N2>
        N3 <DO <%TRACE-MSG "====> AT NODE: " 'N3>
               <COND [T <&STORE (CLASS VOLUME)>
                        <&STORE (VOLUME-TYPE HOLE-IN-GROUND)>
                        <&PP-CLOSE>
                        <(UNFLAG 'WORD4 'SUSPENDED) (%REMPROP-BB '
                         WORD4 'STATUS '-OLD-STATUS) (FLAG 'WORD4 '
                         TERMINATED) (%PUT-BB 'WORD4 'STATUS '
                         TERMINATED) (NOTE 'TERMINATED) (%TRACE-MSG '
                         TERMINATED " EXPERT: " 'WORD4) (ADIEU)>]>>
        N4 <DO <%TRACE-MSG "====> AT NODE: " 'N4>
               <COND [T <&STORE (CLASS FOOD)>
                        <&STORE (FOOD-TYPE FRUIT)>
                        <&STORE (FRUIT-TYPE PART-OF)>
                        <&PP-CLOSE>
                        <(UNFLAG 'WORD4 'SUSPENDED) (%REMPROP-BB '
                         WORD4 'STATUS '-OLD-STATUS) (FLAG 'WORD4 '
                         TERMINATED) (%PUT-BB 'WORD4 'STATUS '
                         TERMINATED) (NOTE 'TERMINATED) (%TRACE-MSG '
                         TERMINATED " EXPERT: " 'WORD4) (ADIEU)>]>>
        N5 <%TRACE-MSG "====> AT NODE: " 'N5>
```

```
EVAL: (%SCAN 'CONCEPT1)

(*PICTURE* CONCEPT2)
(*CONCEPT-CLASS* TIME)
(*CONCEPT-TYPE* SETTING)

VALUE:    CONCEPT1
```

The %scan function lists the facts in a concept bin -- this concept bin contains the three items common to setting bins.

```
EVAL:  (%SCAN 'CONCEPT2)

(*DESCRIPTOR* (TYPE RANGE) WORD3)
(*DESCRIPTOR* (CLASS TIME) WORD3)
(*DESCRIPTOR* DETERMINATE WORD2)
(*CONCEPT-CLASS* TIME)
(*CONCEPT-TYPE* PICTURE)

VALUE:    CONCEPT2
```

This picture bin contains a description of "the morning" including that it is a determined range (timeless duration) of time.

"the deep pit"

Execution of "the deep pit" is an interesting example of the complexity of control flow in the Word Expert Parser. Since both "deep" and "pit" have more than a single sense (implemented!), they must exchange pieces of information about themselves before either can succeed at determining its sense.

```
EVAL:  (%SENTENCE: THE DEEP PIT)
CONTROL STATE: NEW-SENTENCE
VALUE:    T

EVAL:  (%PARSE)

    NEW SENTENCE:   THE DEEP PIT
    WORD NAMES:     WORD1 WORD2 WORD3

EXECUTING 'THE' EXPERT: WORD1
*****TRANSLATING SENSE EXPERT FOR WORD1
**EXPERT COMPILE TIME: 56
**OPTIMIZATION TIME: 1237
*****EXPERT FOR WORD1 TRANSLATED
====> AT NODE: N1
NEW CONCEPT: CONCEPT1   TYPE: PICTURE
CONTROL STATE: PICTURE-CONSTRUCTION
EXPERT TERMINATED: WORD1
```

The "the" expert runs as always!

```
EXECUTING 'DEEP' EXPERT: WORD2
*****TRANSLATING SENSE EXPERT FOR WORD2
**EXPERT COMPILE TIME: 281
**OPTIMIZATION TIME: 5808
*****EXPERT FOR WORD2 TRANSLATED
====> AT NODE: NO
====> AT NODE: N1
EXPERT SUSPENDED: WORD2
```

The "deep" expert runs until it needs to know something of the expert to its right. It suspends execution until "pit" has suspended or terminated. It makes no expectations or constraints, giving "pit" the freedom to be anything.

```
EXECUTING 'PIT' EXPERT: WORD3
*****TRANSLATING SENSE EXPERT FOR WORD3
**EXPERT COMPILE TIME: 307
**OPTIMIZATION TIME: 5852
*****EXPERT FOR WORD3 TRANSLATED
----> AT NODE: N0
----> AT NODE: N1
----> AT NODE: N2
   **RESTART DEMON FOR WORD2 TRIGGERED:
   *    (*EXPERT-STATE* WORD3 SUSPENDED)
   **EXPERT FOR WORD2 QUEUED FIRST
EXPERT SUSPENDED: WORD3
```

"Pit" now runs, but cannot discriminate between "fruit pit" and "hole in the ground". It suspends itself until something happens to help it continue. Since "pit" has now executed, "deep" is reawakened.

```
EXECUTING 'DEEP' EXPERT: WORD2
----> AT NODE: N1
----> AT NODE: N4
   **RESTART DEMON FOR WORD3 TRIGGERED:
   *    (*EXPERT-STATE* WORD2 SUSPENDED)
   **EXPERT FOR WORD3 QUEUED LAST
EXPERT SUSPENDED: WORD2
```

The "deep" expert now says: "I gave the word to my right a chance to be what it wanted and it had no idea. Now I'll constrain it to be something that I can describe -- a hole in the ground or a person." At this point, therefore, "deep" constrains "pit" and suspends itself until "pit" terminates. Since "pit" was waiting for anything, it now queues up.

```
EXECUTING 'PIT' EXPERT: WORD3
----> AT NODE: N2
----> AT NODE: N3
CONTROL STATE: UNKNOWN
   **RESTART DEMON FOR WORD2 TRIGGERED:
   *    (*EXPERT-STATE* WORD3 TERMINATED)
   **EXPERT FOR WORD2 QUEUED FIRST
EXPERT TERMINATED: WORD3
```

"Pit" now runs for the last time, its constraints focusing on the discrimination. Since "pit" terminates, the restart demon for "deep" is triggered, and "deep" is queued.

```
EXECUTING 'DEEP' EXPERT: WORD2
----> AT NODE: N1
----> AT NODE: N3
----> AT NODE: N2
EXPERT TERMINATED: WORD2
CONCEPT CLOSED: CONCEPT1

VALUE:    T
```

Finally, "deep" runs for the last time, terminating with the sense of "large volume". Although "pit" tried to close its concept bin, "deep" still had not terminated, and so a demon was planted to close the concept when "deep" finished. Thus, the picture concept is now closed.

```
EVAL: (%SCAN 'CONCEPT1)

(*DESCRIPTOR* (SIZE-DIMENSION DEPTH) WORD2)
(*DESCRIPTOR* (SIZE LARGE) WORD2)
(*DESCRIPTOR* (VOLUME-TYPE HOLE-IN-GROUND) WORD3)
(*DESCRIPTOR* (CLASS VOLUME) WORD3)
(*DESCRIPTOR* DETERMINATE WORD1)
(*CONCEPT-CLASS* VOLUME)
(*CONCEPT-TYPE* PICTURE)

VALUE:    CONCEPT1
```

The concept bin for "the deep pit" contains a description of a large volume of air in the ground.

Summary and conclusions.- We have mapped out a theory of organization and control for a meaning-based language understanding system. In this theory, words, rather than rules, are the units of knowledge, and assume the form of procedural entities which execute as generator-like coroutines. Parsing a sentence in context demands a control environment in which these experts can ask questions of each other, forward hints and suggestions to each other, and suspend. Our theory is a cognitive theory of both language representation and parser control.

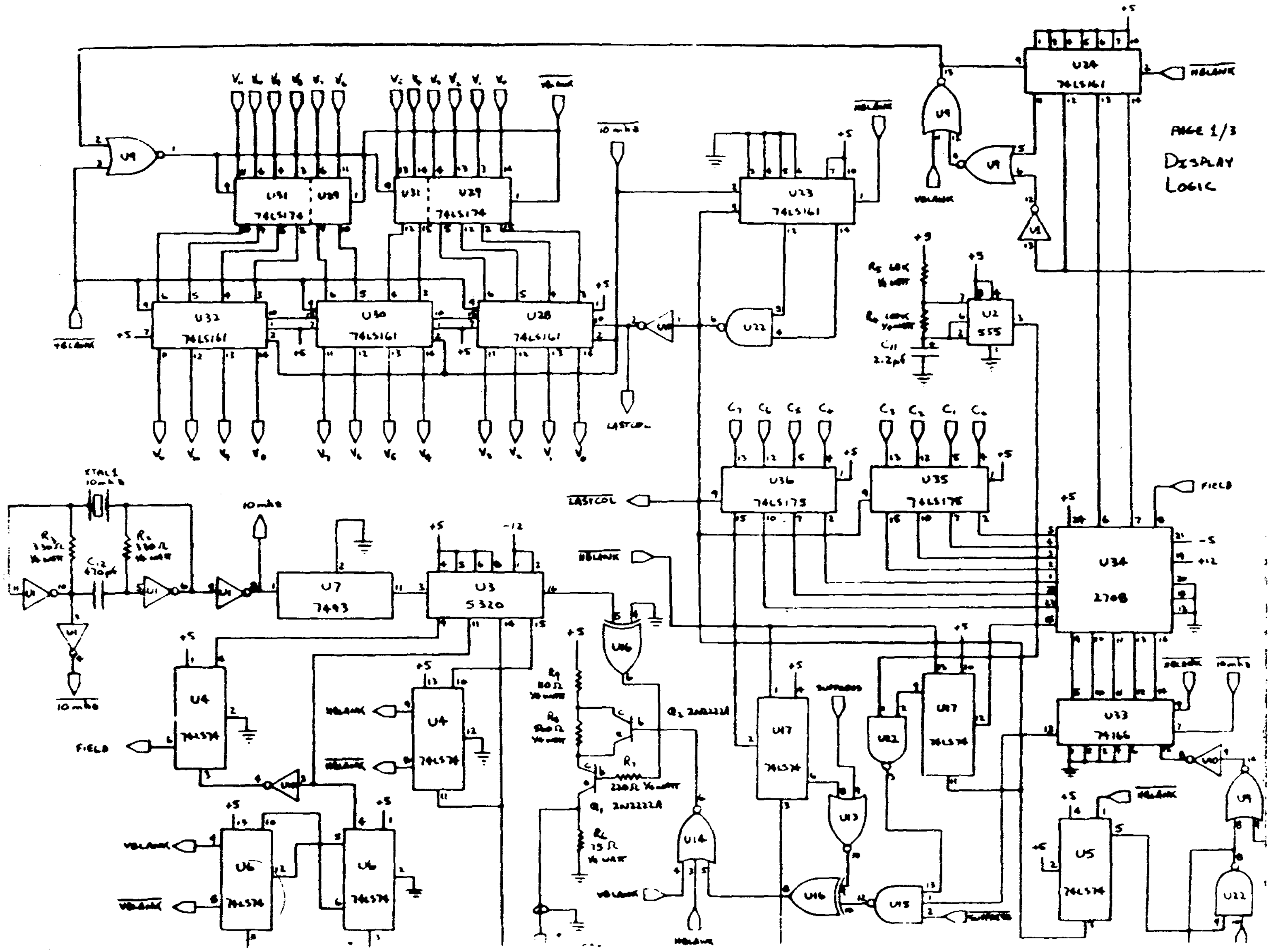
We are using this type of work in approaching the problems of the second topic I would like to discuss; that is, SADD for semi-automatic digital design.

We try to provide a semi-informed digital designer with a scratch pad capability in the initial phases of the design.

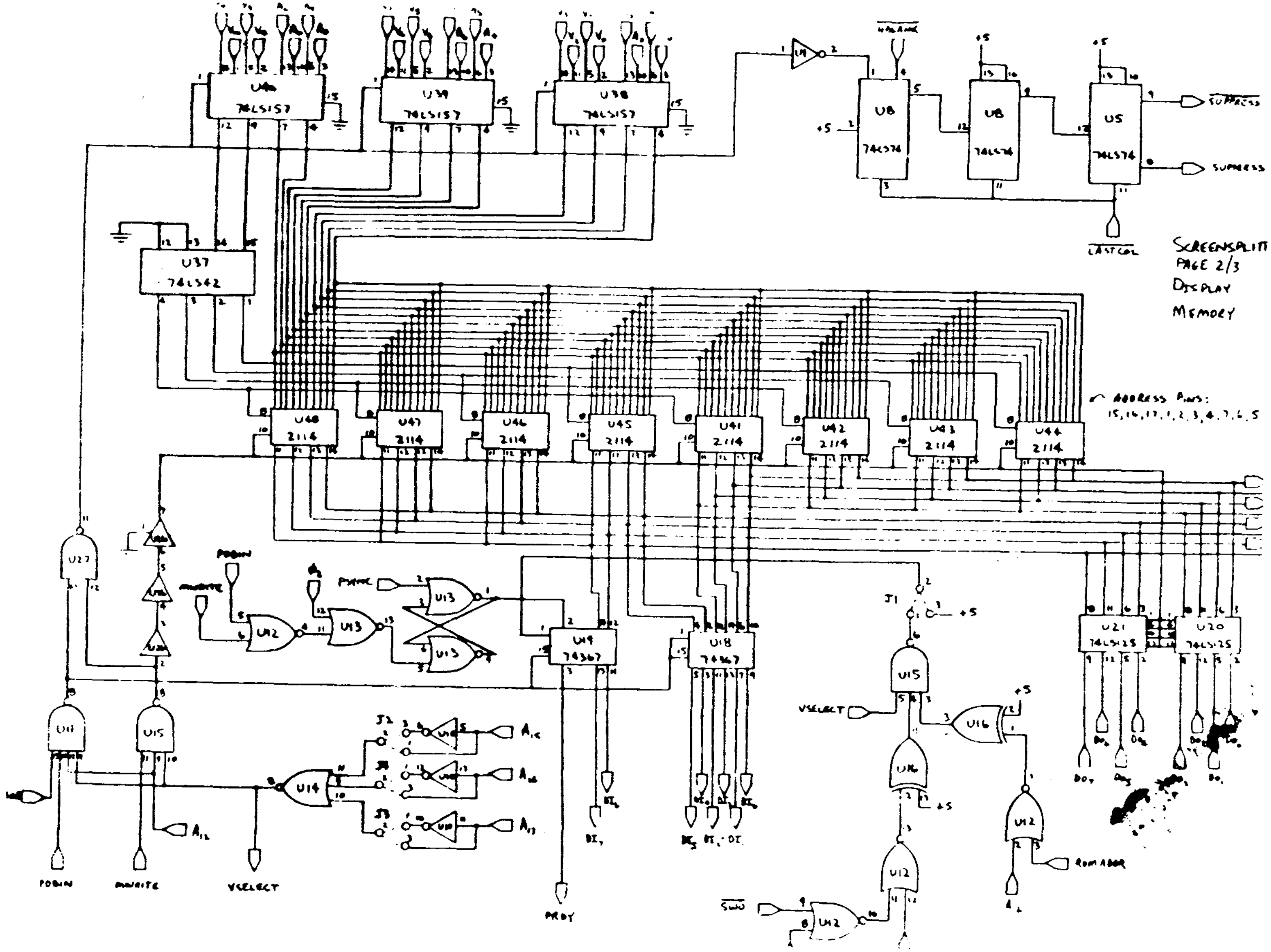
I like to show this as an application of the notion word expert parser.

VIRTUES OF SENSE NET PARSING

1. ALLOWS MODULAR GROWTH OF LANGUAGE KNOWLEDGE (NUMEROUS RESEARCHERS CAN MAKE RELATIVELY INDEPENDENT CONTRIBUTIONS)
2. KNOWLEDGE IS GLOBALLY DISTRIBUTED, LOCALLY CONCENTRATED (IT IS RELATIVELY TRANSPARENT, UNLIKE GRAMMAR'S)
3. KNOWLEDGE IS REDUNDANT (COMPREHENSION NEVER FAILS TOTALLY)
4. KNOWLEDGE INTERACTS MODULARLY, USING FAIRLY SIMPLE CONTROL STRUCTURES
5. SENSE NETS SUGGEST FORMAT FOR LEARNING LANGUAGE (THEY PROVIDE STRUCTURES THAT ARE RELATIVELY EASY TO AUGMENT)
6. IT LEADS TO SOME DIRECT IDEAS ABOUT CONTEXTUAL INTERACTION AMONG WORDS (CONSTRAINT FORWARDING, MODEL CONTEXT INTERACTION)
7. ITS FORM DOES NOT CONSTRAIN ONE'S AMBITIONS AS MUCH AS GRAMMAR'S (IT IS EASIER TO INTERACT WITH OTHER MODEL COMPONENTS)
8. IT IS EASILY SEGMENTABLE (SENSE NETS CAN BE STORED ON RANDOM ACCESS MEDIUM, ASSEMBLED AS REQUIRED)
9. IT ACCOUNTS FOR MANY IDIOSYNCRASIES OF LANGUAGE UNDER A UNIFORM DATA STRUCTURE (WORD-SPECIFIC CONSTRUCTIONS ARE NATURAL TO ENCODE)
10. IT HAS SOME INTUITIVE APPEAL AS BEING CLOSE TO HOW HUMANS (WELL, OK... HOW I) PARSE AND LEARN LANGUAGE



Just to show you that we are talking about a real system, this slide presents one-third of a schematic of a TV text display. The next slide is page 2 of the design.


SCREENSPLIT
PAGE 2/3
DISPLAY
MEMORY
ADDRESS PINS:
15,16,17,1,2,3,4,7,6,5
U37
74LS42
74LS157
74LS74
2114
74367
74LS125
SUPPRESS
LASTCOL
PRDY
VSELECT
ROM ADDR

We described the circuit by these 41 sentences which are the input to our current model. We spell everything out. The model has about 9 conceptual categories; e.g., it knows about clocks, latches, buffering, counting, combinational circuits. We call these things frames, a la Minsky.

Gradually, we get an overall framework.

SCREENSPLITTER DESCRIPTION

1. THE DISPLAY CHARACTER COUNTER (DCC) COUNTS FROM 0 TO 3519.

THE DCC-BASE REGISTER (DCCBR) RESETS DURING VERTICAL BLANK (VBLANK).

THE SCAN LINE COUNTER (SLC) COUNTS FROM 15 TO 4.

4. THE SLC INCREMENTS WHEN THE HORIZONTAL BLANK (HBLANK) ENDS.
5. THE SLC RESETS WHEN VBLANK FALLS.
6. WHEN THE COUNT OF THE SLC EQUALS 4, THE COUNT OF THE DCC CAPTURED-IN THE LOAD OF THE DCCBR WHEN HBLANK BEGINS.
7. THE DCC CAPTURED-FROM THE DCCBR WHEN HBLANK BEGINS.
8. EACH-TIME THE COUNT OF THE PIXEL COUNTER (PC) EQUALS 5 ENDS, THE DCC INCREMENTS.
9. THE PIXEL COUNTER (PC) COUNTS FROM 0 TO 5.
10. THE PC INCREMENTS EACH-TIME THE OUTPUT OF THE MASTER CLOCK (MC) FALLS.
11. A COUNT OF THE PIXEL COUNTER (PC) EQUAL TO 5 CALLED LASTCOL.
12. THE COUNT OF THE PC RESETS TO 0 DURING HBLANK.
13. THE COUNT OF THE DCC CAPTURED-IN THE LOW-ORDER BITS OF THE ADDRESS OF THE DISPLAY MEMORY (DM).
14. THE MASTER CLOCK (MC) DUTY-CYCLE IS SQUARE-WAVE.
15. THE MC FREQUENCY IS 10MHZ.
16. THE MASTER CLOCK (MC) DIVIDED-BY 8 AND CAPTURED-IN THE CLOCKED-INPUT OF THE SYNC-GENERATOR (SG).
17. THE SYNC-GENERATOR (SG) PRODUCES FIELD.
18. THE SG PRODUCES HBLANK.
19. THE SG PRODUCES VBLANK.
20. THE SG PRODUCES CSYNC.
21. THE 2 LOW-ORDER BITS OF THE READ OF THE SLC SENT-TO LINE 2 AND LINE 3 OF THE ADDRESS OF THE CHARACTER GENERATOR MEMORY (CG).

FIELD SENT-TO LINE 1 OF THE ADDRESS OF CG.

CG CHIP IS EPROM-2708.

24. THE FETCHED CHARACTER REGISTER (FCL) WIDTH IS 8.
25. THE READ OF THE DM LATCHED-INTO THE FCL EACH-TIME LASTCOL FALLS.
26. THE VALUE OF THE FCL SENT-TO THE HIGH-ORDER 7 LINES OF THE ADDRESS OF THE CG.
27. THE 5 LOW-ORDER LINES OF THE READ OF THE CHARACTER GENERATOR MEMORY (CG) LOADED-INTO THE SHIFTER (SR) WHEN LASTCOL FALLS AND WHEN THE LEFT MARGIN SUPPRESSOR SWITCH (LMS) IS OFF.
28. BIT 6 OF THE LOAD OF THE SR LOADED-WITH 0 WHEN LASTCOL FALLS AND WHEN THE LMS IS OFF.
29. THE LMS TURNED-ON BY HBLANK.
30. THE LMS TURNED-OFF WHEN LASTCOL FALLS.
31. THE DISPLAY MEMORY (DM) WORDS ARE 4096.
32. THE DM WORD-WIDTH IS 8.
33. THE HOST ADDRESS BUSS (HSA) SENT-TO THE LOW-ORDER ADDRESS OF THE DM EITHER WHEN A MEMORY READ REQUEST (MR) OCCURS OR WHEN A MEMORY WRITE REQUEST (WR) OCCURS.
34. THE OUTPUT VIDEO (VID) OF THE SR IS-INVERTED WHEN FIGURE GROUND LOGIC (FGL) IS ON TO-BECOME FVID.
35. THE FVID SUPPRESSED EITHER BY THE OUTPUT OF THE WINK CLOCK (WC) OR BY SUPPRESSOR LOGIC (SL) TO-BECOME OUTVID.
36. OUTVID SUPPRESSED EITHER BY HBLANK OR BY VBLANK TO-BECOME TVVID.
37. THE TVVID SENT-TO THE TV INTERFACE LOGIC (TIL).
38. THE FGL LOADED-FROM THE HIGH-ORDER BIT OF THE FCL EACH-TIME LASTCOL FALLS AND WHEN LMS IS ON.

THE WC DUTY-CYCLE IS SQUARE-WAVE.

40. THE WC FREQUENCY IS 0.5HZ.
41. THE SL ENABLED EITHER BY A MR OR BY MW, AND DELAYED-FOR 3 LASTCOL PERIODS.

DESIGN OVERVIEW

- SPECIFICATION ACQUISITION
- SPECIFICATION DEDUCTION
- CIRCUIT SELECTION
- CIRCUIT IMPLEMENTATION
- CIRCUIT SIMULATION

To give an example of how SADD works, the parser is a very abbreviated work of "expert parser." All individual words are represented by procedures. The sentence G1..... builds the small data base G1-G25.

The following slides show the kinds of layouts that get constructed.

PARSING

INITIAL SENTENCE

WHEN THE COUNT OF THE SLC EQUALS 4, THE COUNT OF THE DCC CAPTURED-IN THE LOAD OF THE DCC-BASE REGISTER (DCCBR) WHEN THE HORIZONTAL BLANK (HBLANK) BEGINS.

OBJECT TYPE FINAL PARSE

<conditional> *COMMA* <signal-obj> CAPTURED-IN <signal-obj> <conditional> *PERIOD*

FINAL PARSE

G1 *COMMA* G7 CAPTURED-IN G13 G20 *PERIOD*

G1 -
G2 - (G1 TYPE CONDITION)
G3 - (G1 SIGNAL-MNEM DS4)
G4 - (G1 FUNCTION-MNEM SLC)
G5 - (G1 CONDITION HIGH)
G6 - (G1 FUNCTION-TYPE COUNTER)

G7 -
G8 - (G7 TYPE SIGNAL)
G9 - (G7 SIGNAL-MNEM COUNT)
G10 - (G7 FUNCTION-MNEM DCC)
G11 - (G7 CONDITION HIGH)
G12 - (G7 FUNCTION-TYPE COUNTER)

G13 -
G14 - (G13 TYPE SIGNAL)
G15 - (G13 SIGNAL-MNEM LOAD)
G16 - (G13 FUNCTION-MNEM DCCBR)
G17 - (G13 CONDITION HIGH)
G18 - (G13 FUNCTION-NAME (DCC-BASE REGISTER))
G19 - (G13 FUNCTION-TYPE REGISTER)

G20 -
G21 - (G20 TYPE CONDITION)
G22 - (G20 SIGNAL-MNEM HBLANK)
G23 - (G20 FUNCTION-MNEM UNKNOWN)
G24 - (G20 CONDITION UP)
G25 - (G20 SIGNAL-NAME (HORIZONTAL BLANK))

VERB EFFECT ON MODEL

Verify G7 is a signal-object
Verify G18 is a signal-object
Connect G13 to G7 under conditions G1 and G20

PARSING DIRECTIVES FOR MANIPULATING THE MODEL

1. SPECIFY A FUNCTION

   THE DISPLAY CHARACTER COUNTER (DCC) COUNTS FROM 0 TO 3519.

2. ASSIGN A VALUE TO A FUNCTION'S ASPECT

   THE WINK CLOCK (WC) FREQUENCY IS 0.5 HZ.

3. MAKE A CONNECTION

   THE LOAD OF THE DCC CAPTURED-FROM THE DCCBR WHEN HBLANK BEGINS.
   THE DCCBR RESETS DURING VBLANK.

4. DEFINE A CONCEPTUAL SIGNAL

5. DEFINE THE SOURCE OF A CONCEPTUAL SIGNAL

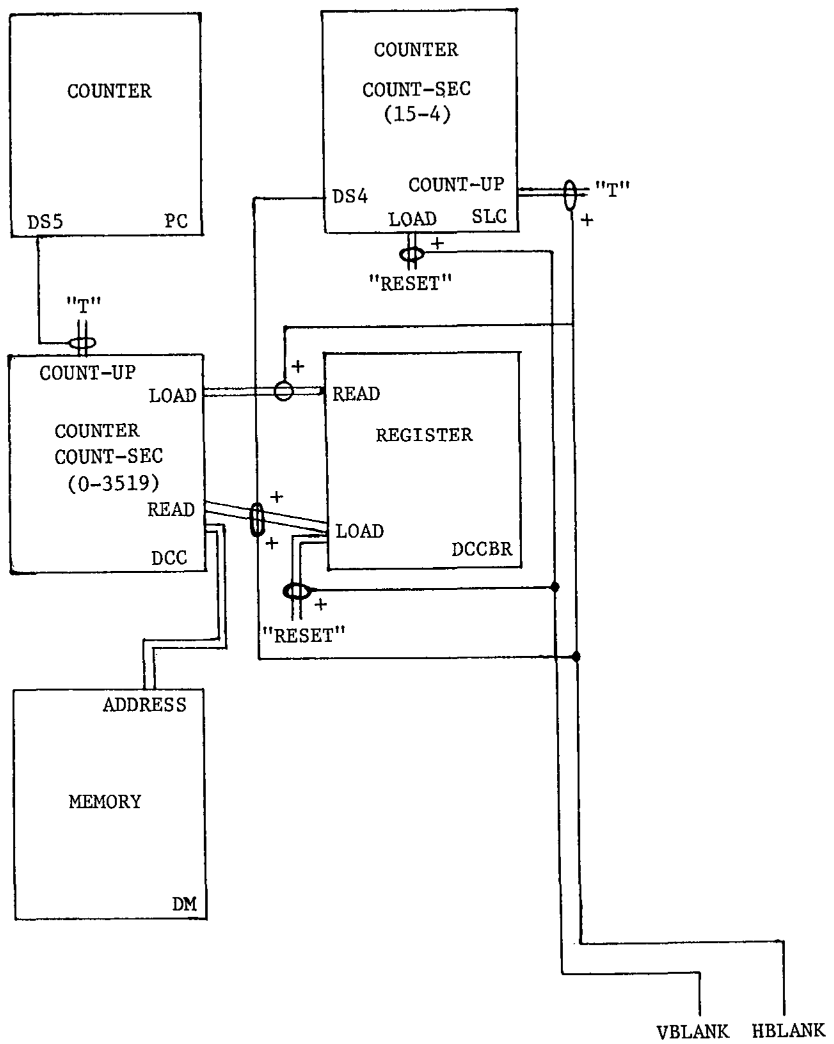
   THE SYNC GENERATOR (SG) PRODUCES VBLANK.

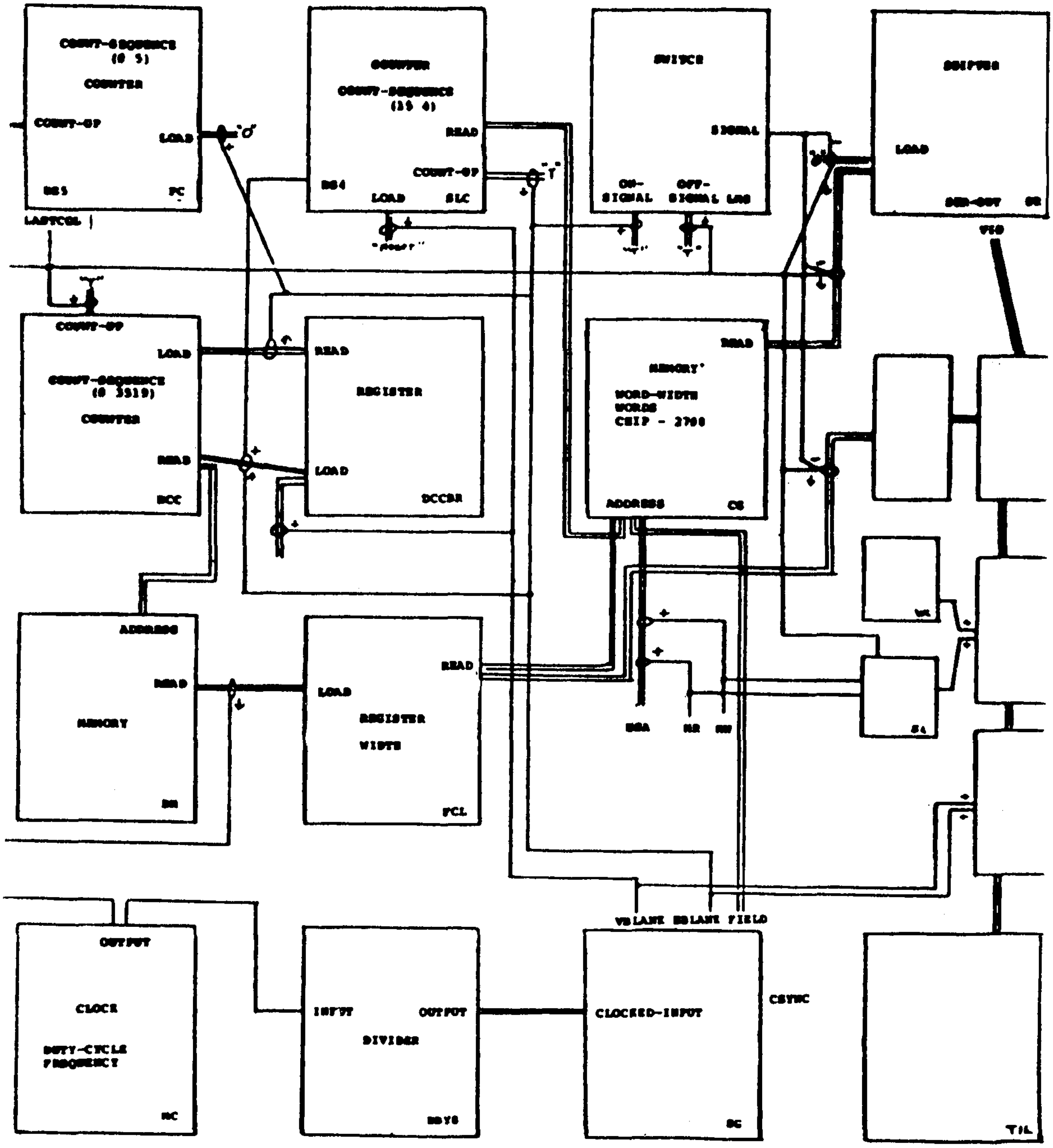
1. THE DISPLAY CHARACTER COUNTER (DCC) COUNTS FROM 0 TO 3519.
2. THE DCC-BASE REGISTER (DCCBR) RESETS DURING VERTICAL BLANK (VBLANK).
3. THE SCAN LINE COUNTER (SLC) COUNTS FROM 15 TO 4.
4. THE SLC INCREMENTS WHEN THE HORIZONTAL BLANK (HBLANK) ENDS.
5. THE SLC RESETS WHEN VBLANK FALLS.
6. WHEN THE COUNT OF THE SLC EQUALS 4, THE COUNT OF THE DCC CAPTURED-IN THE LOAD OF THE DCCBR WHEN HBLANK BEGINS.

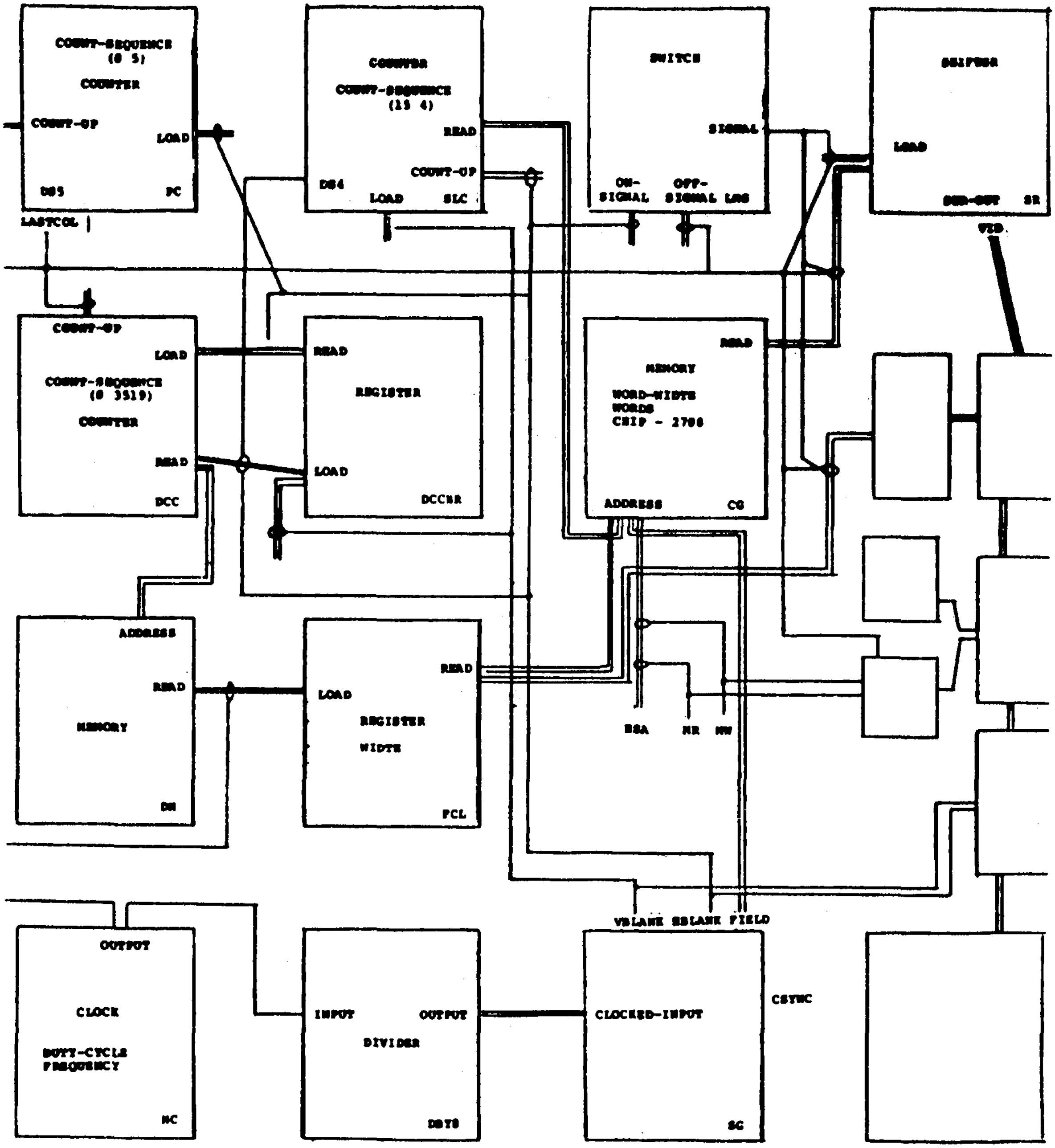
7. THE DCC CAPTURED-FROM THE DCCBR WHEN HBLANK BEGINS.

8. EACH-TIME THE COUNT OF THE PIXEL COUNTER (PC) EQUALS 5 ENDS, THE DCC INCREMENTS.

13. THE COUNT OF THE DCC CAPTURED-IN THE LOW-ORDER BITS OF THE ADDRESS OF THE DISPLAY MEMORY (DM).

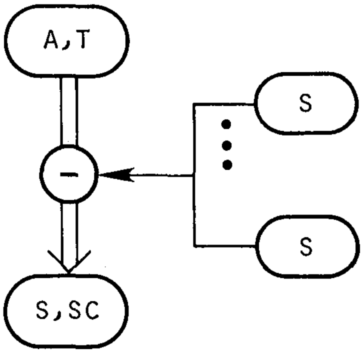



COUNTER
COUNT-UP
LOAD
LASTCOL
READ
LOAD
SIGNAL
SWITCH
LOAD
COUNT-UP
LOAD
READ
REGISTER
LOAD
COUNTER
READ
MEMORY
WORD-WIDTH
WORDS
ADDRESS
ADDRESS
READ
MEMORY
READ
LOAD
REGISTER
WIDTH
FCL
VBLANK HBLANK FIELD
OUTPUT
CLOCK
DUTY-CYCLE
FREQUENCY
INPUT
OUTPUT
DIVIDER
CLOCKED-INPUT
CSYNC

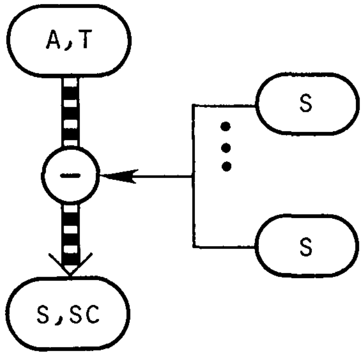

COUNT-SEQUENCE
(0 5)
COUNTER
COUNT-UP
LOAD
PC
LASTCOL
COUNTER
COUNT-SEQUENCE
(15 4)
READ
COUNT-UP
DB4
LOAD
SLC
SWITCH
SIGNAL
ON-
SIGNAL
OFF-
SIGNAL
SHIFTER
LOAD
SR
VID
COUNT-UP
LOAD
COUNT-SEQUENCE
(0 3519)
COUNTER
READ
DCC
READ
REGISTER
LOAD
READ
MEMORY
WORD-WIDTH
WORDS
ADDRESS
CG
ADDRESS
READ
MEMORY
DM
READ
LOAD
REGISTER
WIDTH
PCL
MR
MW
VBLANK
HBLANK
FIELD
OUTPUT
CLOCK
DUTY-CYCLE
FREQUENCY
MC
INPUT
OUTPUT
DIVIDER
CLOCKED-INPUT
CSYNC
SG

The common sense algorithm is a general purpose theory of how humans perceive causality. This theory developed several links as shown here. We distinguish one-time causality vs. continuous causality. The primitives that enter are action, state, and state change. States enable actions. Shown here are several representations of causal links.

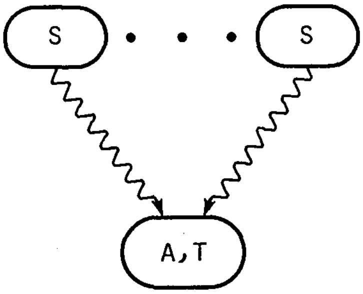
CAUSAL LINKS



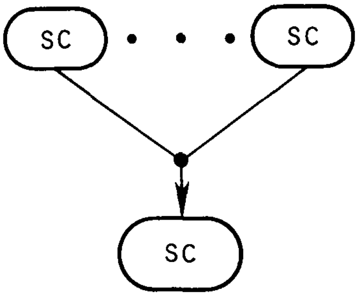
COMMON SENSE
CAUSAL KNOWLEDGE
(CSA)



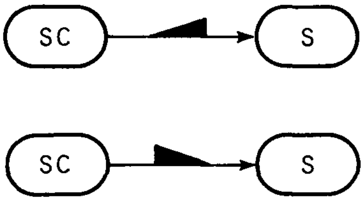
ENABLEMENT

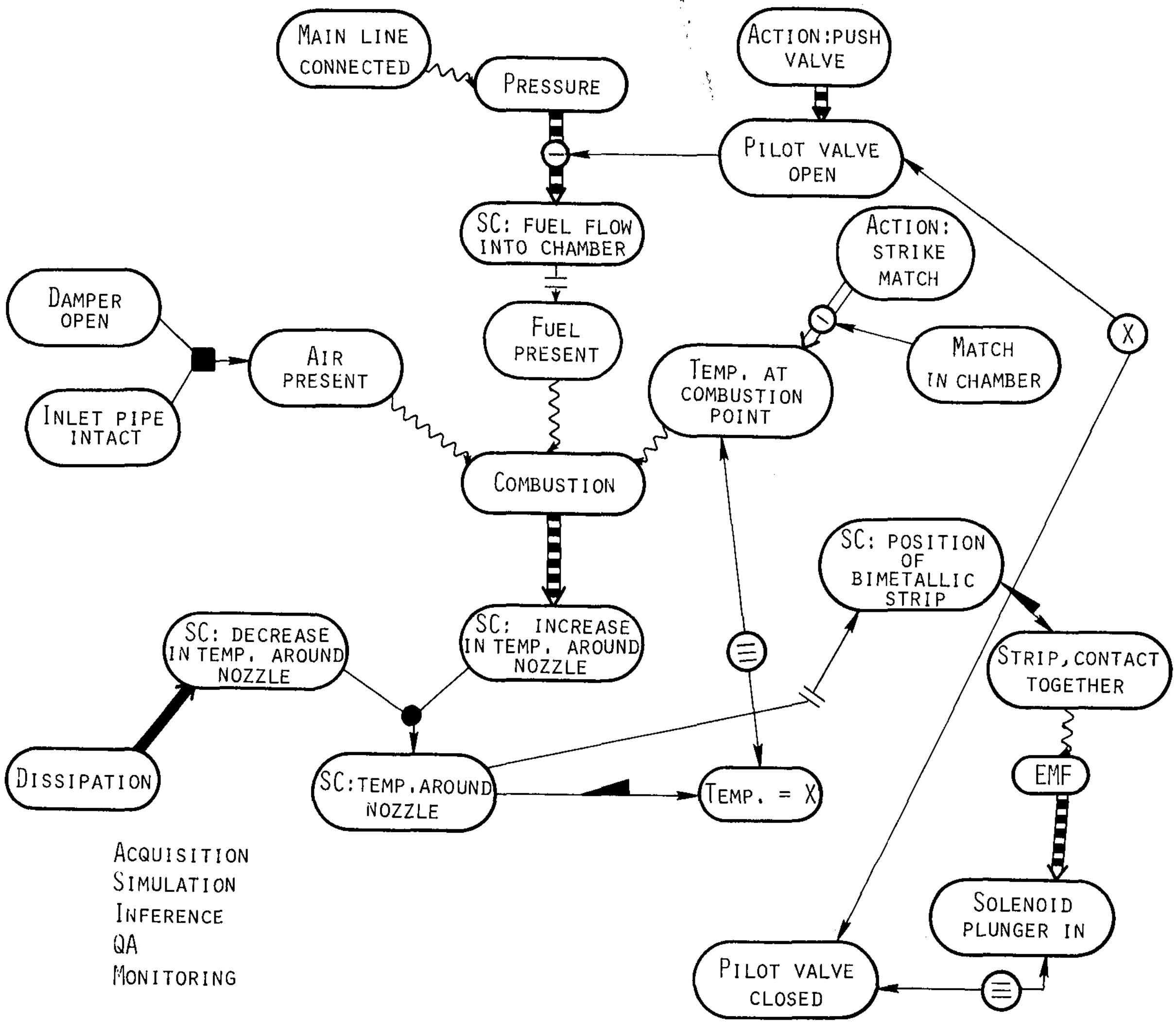


CONFLUENCE



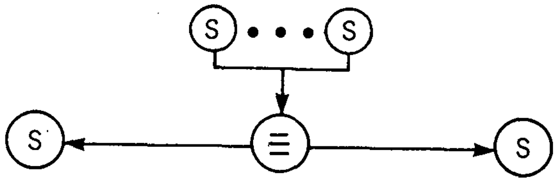
THRESHHOLD



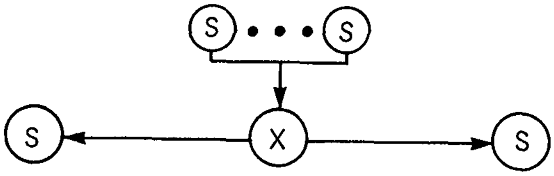


This is an approximation of the workings of a furnace. We have here everything all spelled out with a collection of rules with all actions and states and state changes. We can represent fairly complex causal models this way.

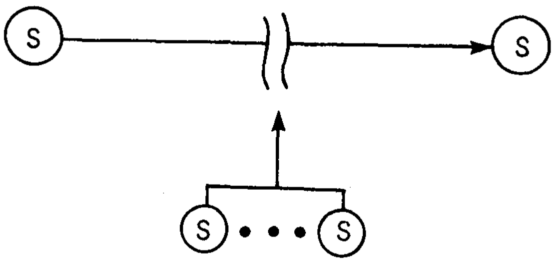
STATE EQUIVALENCE



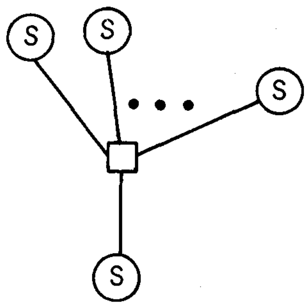
STATE ANTAGONISM (FEEDBACK LOOPS)



STATE COUPLING



COMPOUND STATE



In monitoring a complex physical site which should become aware of itself, we came up with this project CAM (Causal Monitor Project). There is the obvious need for causal models (monitors) with a good repertoire of abstract causality and the ability to interact with the experts as they see a particular site. These are the things we want the system to do.

CAM GOALS

1. CONTINUOUSLY SENSE AND SYMBOLICALLY MONITOR ALL SENSORS IN A WAY THAT IS APPROPRIATE IN THE CURRENT OPERATING CONTEXT.

2. CONTINUALLY VERIFY THAT CAUSALLY RELATED SENSOR GROUPS OBEY SYMBOLICALLY EXPRESSED RULES ABOUT THEIR INTER-RELATIONSHIP.

3. BE AWARE OF HUMAN OPERATOR'S EXPRESSED INTENT AND ADJUST EXPECTATIONS AND CAUSAL RELATIONS ACCORDINGLY.

4. KNOW ABOUT COMPONENT AND SENSOR FAILURE MODES, THEIR PRE-CURSORS, INDICATORS, PROBABLE CAUSES AND CORRECTIVE PROCE-DURES AND POLICIES.

5. KNOW ABOUT STANDARD "MANEUVERS" EXPRESSED AS ACTION SEQUENCES WITH STEP-WISE EXPECTATIONS AND CONFIRMATION.

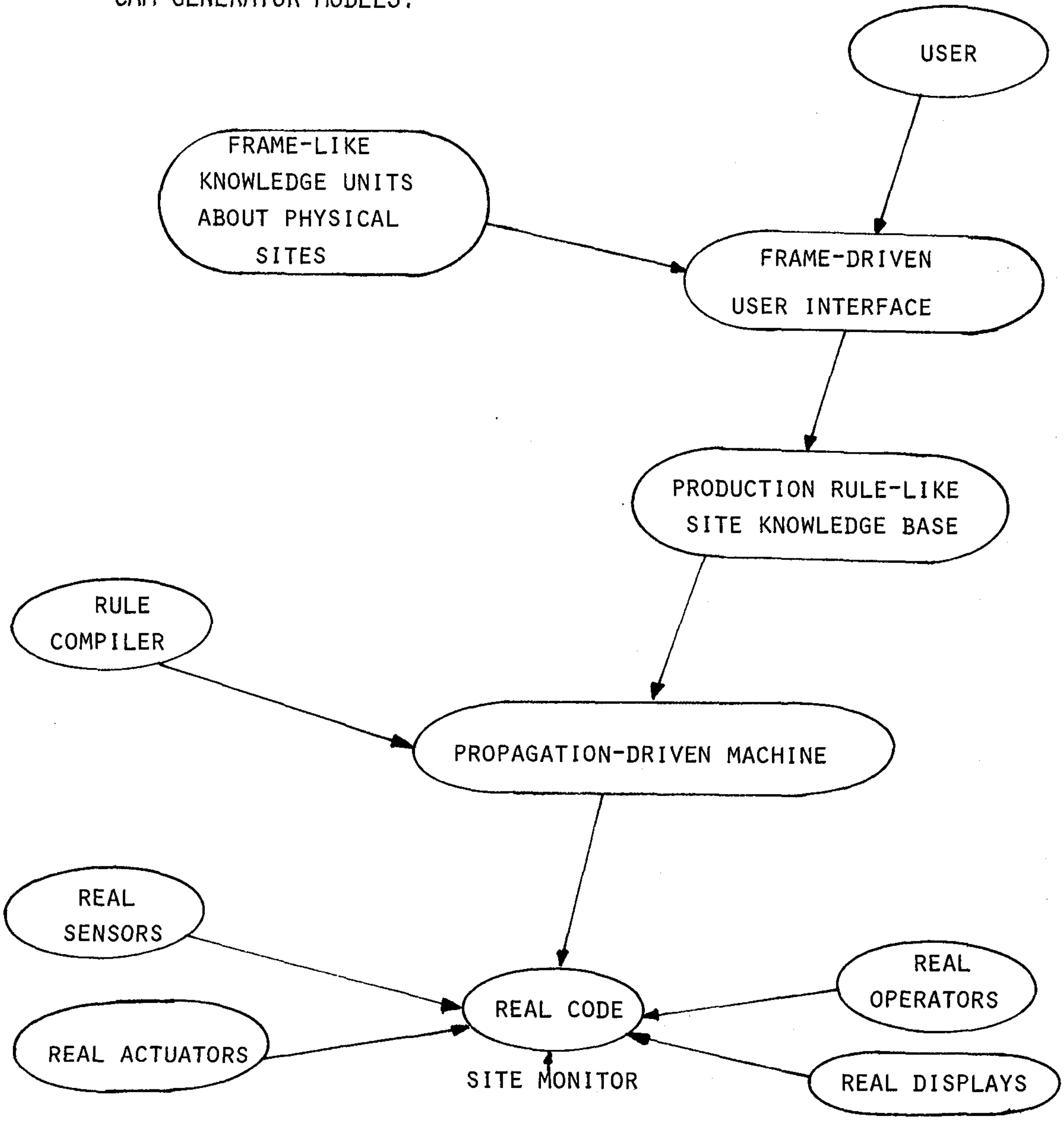
6. CONTINUOUSLY SYNTHESIZE ALL HIGH LEVEL SYSTEM ASPECTS AND SELECT WHICH INFORMATION TO DISPLAY AND IN WHAT FORMAT (INTELLIGENT SCREEN CONTROLLERS).

7. CONTAIN ON-LINE POLICY INFORMATION FOR CONTINGENCY SITUATIONS.

REAL-TIME CAUSAL MONITORS

FOR COMPLEX PHYSICAL SITES

CAM GENERATOR MODELS:



PHYSICAL SITE FRAMES

SENSOR	NAME, PHYSICAL UNITS SENSED, SENSING RATE IN CONTEXT, LIMITS IN CONTEXT
ACTUATOR	NAME, UNITS CONTROLLED, DRIVER PROCEDURE, MEASURABLES (EXPECTATIONS)
COMPONENT	NAME, ASSOCIATED SENSORS, ASSOCIATED ACTUATORS, ASSOCIATED FAILURE MODES
FAILURE MODE	SENSOR CLUES, PROBABLE CAUSES, APPLICABLE CORRECTIVE MANEUVERS, ASSOCIATED DISPLAY PACKETS
ACTION STEP	RELEVANT ACTUATORS (CLUSTERS) AND MEASURABLES, ASSOCIATED MANEUVERS
MANEUVER	STEP SEQUENCE IN CONTEXT, INTENDED GOAL
OPERATING CONTEXT	GLOBAL SENSOR RELATIONS, APPROPRIATE MANEUVERS, APPROPRIATE DISPLAY PACKETS
DISPLAY PACKETS	SENSOR GROUPS, SCREEN ALLOCATION IN CONTEXT, DISPLAY FORMAT
OPERATOR INTENT	ASSOCIATED MANEUVERS (AND GOALS), ASSOCIATED CONTEXTS

IF

CONTEXT = MORNING POWER UP

AND

PRESSURE • SENSOR 17 (SAMPLE AT 5 SEC) -
PRESSURE • SENSOR 32 (SAMPLE AT 5 SEC)
> 0.5

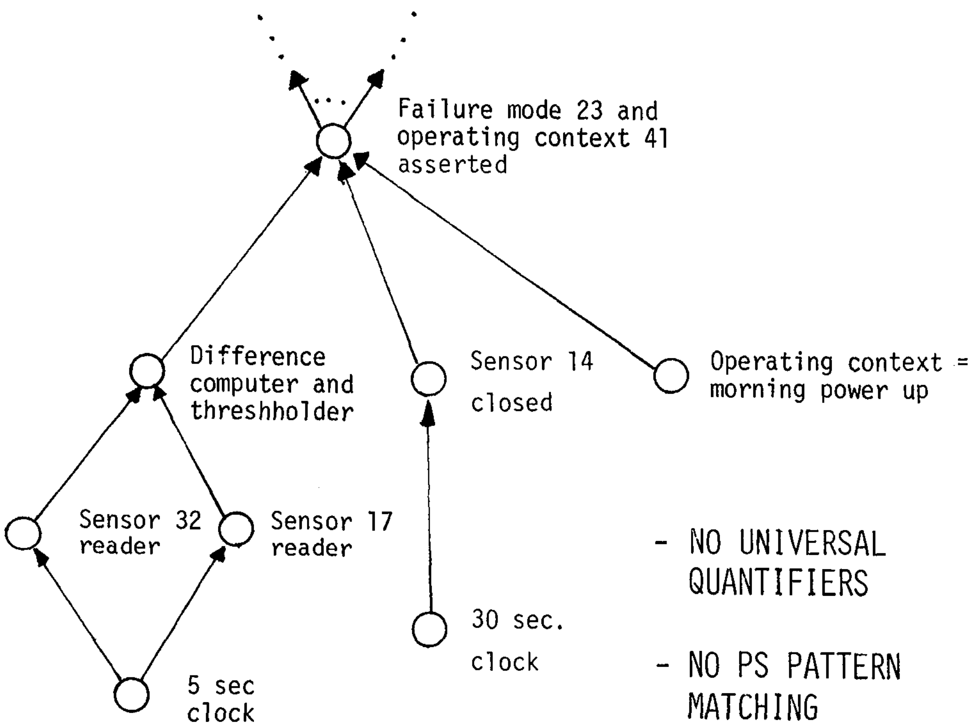
AND

SENSOR 14 = CLOSED (SAMPLE AT 30 SEC)

THEN

ASSERT FAILURE MODE 23

ASSERT OPERATING CONTEXT 41



- NO UNIVERSAL QUANTIFIERS

- NO PS PATTERN MATCHING

The common theme of all our work has been distribution of information. The question is what do we do to back all of this up with real computing. There are not many models around to run in real time. It is particularly bad, e.g., in vision. You can not do everything on conventional machines. You have to make some commitments to parallelism and parallel processing. We had some ideas and worked on a system we call ZMOB.

One Z80A can not do much, but 256 Z80A's can do a lot, doing fairly complex vision algorithms with considerable speed. This is not yet ready for use. This is not a synchronous machine. The ZMOB works like a conveyor belt which will be able to shift about 20 MHz. We hope this will become a community-wide resource at Maryland University.

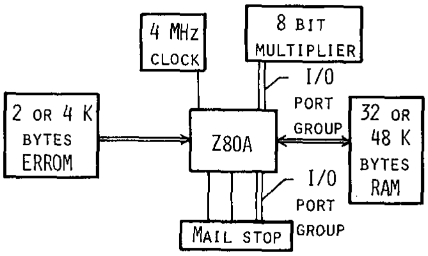
DISTRIBUTION OF COMPUTATION TO MEET REAL-TIME CONSTRAINTS

- THE PDM LATTICE CAN BE PARTITIONED (E.G., SENSOR GROUPS, CLOCKS, TRESHHOLD MEASURERS, FAILURE MODES)

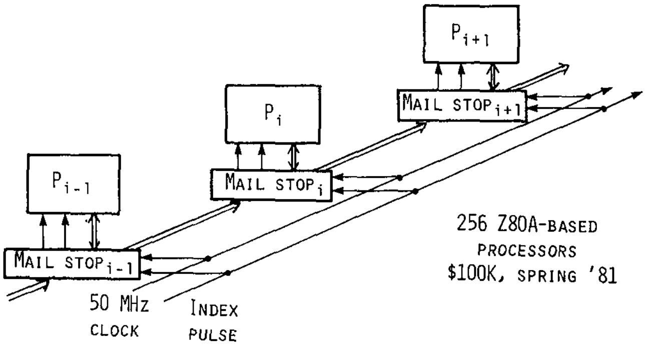
- INDIVIDUAL PROCESSORS NEED ONLY KNOW THEIR ABOVE NODES (AND HENCE WHERE TO SEND THE INFORMATION WHEN A CHANGE OCCURS)

- THE SPECIALIZED TASKS (OPERATOR INTERFACE, DISPLAY HANDLERS) CAN BE PUT IN A SUBSET OF THE PROCESSORS

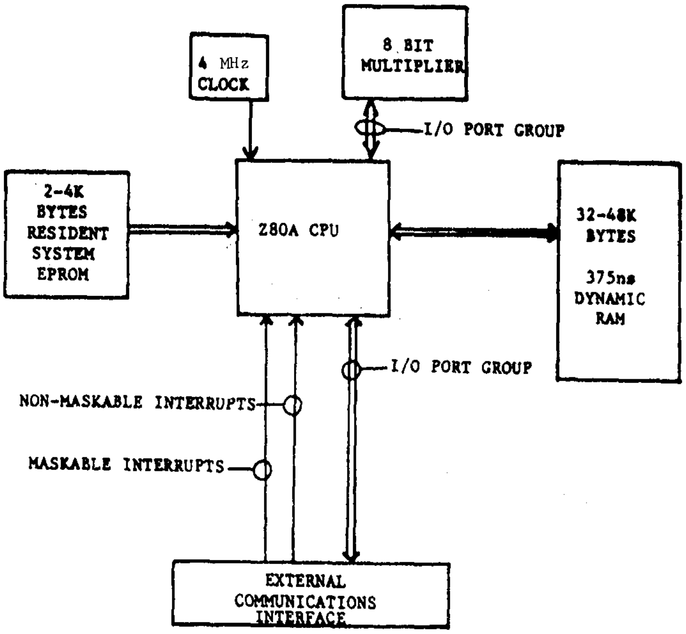
- THE SYSTEM AT LARGE CAN BE AS REDUNDANT AS NECESSARY



EACH PROCESSOR



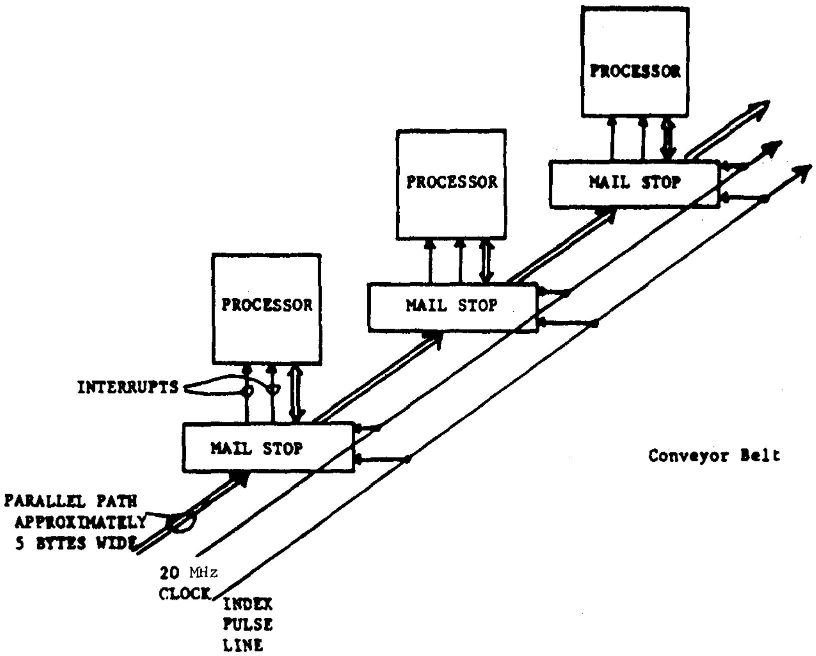
ZMOB

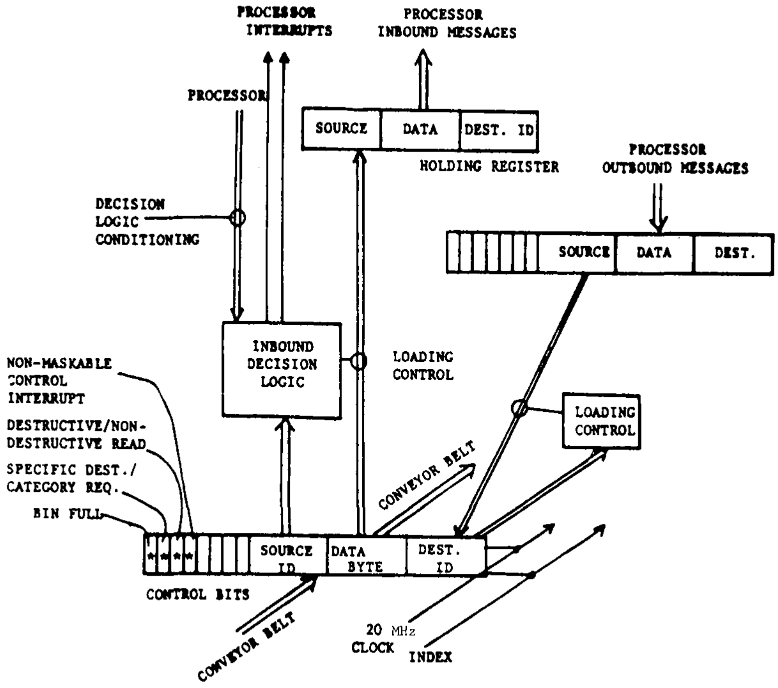


Processor Organization.

Z80A

- 8-BIT ARCHITECTURE
- 65K ADDRESS SPACE
- 256 I/O PORT SPACE
- 4 MHZ CYCLE TIME
- 2 μs TYPICAL INSTRUCTION TIME
- 7 8-BIT WORKING REGISTERS (2 SETS)
- STACK REGISTER
- PROGRAM COUNTER
- IX, IY INDEX REGISTERS
- APPROXIMATELY 150 INSTRUCTIONS
- FAST BLOCK TRANSFER AND SEARCH INSTRUCTIONS
- EXISTING HIGH LEVEL LANGUAGES (PASCAL, LISP)
- APPROXIMATELY $10:





Mail Stop.

THROUGHPUT:

1/3 SEC	9 POINT AVERAGE (8-BIT INTENSITIES) OVER A 512 x 512 IMAGE
1 SEC	SIMPLE EDGE DETECTOR AT EACH POINT OF 512 x 512 IMAGE
100 SEC	COMPLICATED 10 LABEL RELAXATION ALGORITHM ON 512 x 512 x 10 IMAGE
200,000 BYTES/SEC	SWAPPING RATE: INDIVIDUAL PROCESSOR
20-50 MILLION BYTES/SEC	SWAPPING RATE: ZMOB

PROBLEM SOLVING WITH UNCERTAIN KNOWLEDGE

Bruce Buchanan
Stanford University

First, some basics to set the content. Then I will give some specific examples. At the end, I want to indicate the state of the art.

Basically, we like to divide the world of problem solving as shown here.

In AI, we try to deal with symbolic problem solving techniques for which a mathematical technique is not available.

TYPES OF COMPUTING

	NUMERIC	SYMBOLIC
ALGORITHMIC	Computational Mathematics	Information Storage & Retrieval
HEURISTIC	Unprovable Assumptions in Computations	Artificial Intelligence

We believe that we build systems capable of expert level performance. The system represents domain-specific knowledge such as knowledge about geology, medicine, etc. and enables a process in which it uses such knowledge in an understandable line of reasoning. The MYCIN system is one example of such a system.

EXPERT SYSTEM

- CAPABLE OF EXPERT-LEVEL PERFORMANCE
- REPRESENTS DOMAIN-SPECIFIC KNOWLEDGE NATURALLY
- USES KNOWLEDGE IN AN UNDERSTANDABLE LINE OF REASONING

Now we are at the level of building computer programs at the level of assistant.

PROBLEM SOLVING ACTIVITIES

- CLERK - TABULATIONS
- STATISTICIAN _ ASSOCIATIONS
- ASSISTANT - INTERPRETATIONS
- MANAGER - "HARD" DECISIONS
- SCIENTIST - NEW TOOLS
- POLITICIAN - NEW GOALS
- PHILOSOPHER - NEW VALUES

These are ill-structured problems in which we deal with empirical data. The sources of uncertainty are therefore numerous as shown here. You never have a complete set of data in most cases, and you have no guarantee that the data is correct or that your rules of reasoning are complete or correct. In general, the theory is often incomplete for the particular domain of knowledge. You have to use judgemental rules to get at many of these problems.

SOURCES OF UNCERTAINTY

MISSING DATA

ERRONEOUS DATA

MISSING RULES

ERRONEOUS RULES

INCOMPLETE THEORY

We have to rely on redundancy in the data. Often you have overlapping data, overlapping rules, etc. We can rely on the expert heuristics to solve a class of problems. If this can be captured in a computer program, then the computer program will do better than the non-expert. At the moment, we can not capture all but we have to exploit some techniques; e.g., the action strategy enables us to exploit the redundancy to reach correct solutions.

CORRECTIONS FOR UNCERTAINTY

REDUNDANT DATA

REDUNDANT RULES

EXPERTS' HEURISTICS

CAUTIOUS STRATEGY

These are the general issues in building an expert system. It does not need to be a simulation, but it needs to be understandable. How can knowledge be employed in expert problem solving, heuristic search, model of cooperating experts, incremental requirement, etc?

There is no clearcut answer for best control structure. Knowledge acquisition is a bottleneck. We are getting better, e.g., 5-10 years for mass spectrometry to 2-3 years for the MYCIN system, and a few weeks now.

Explanation is essential in an expert system. The person using the system needs to understand how the conclusions have been arrived at. He needs to be able to judge whether he wants to accept or reject the answer.

ISSUES

KNOWLEDGE REPRESENTATION

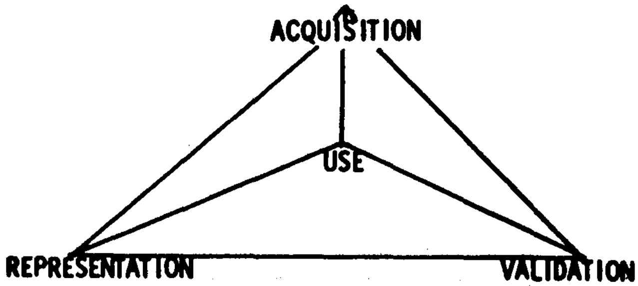
KNOWLEDGE UTILIZATION

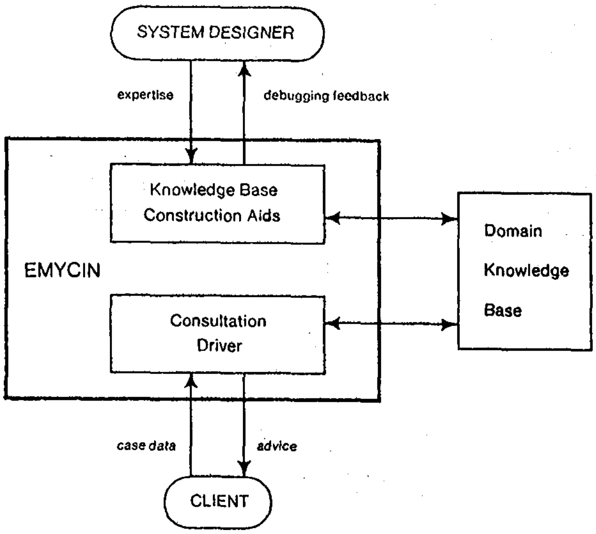
KNOWLEDGE ACQUISITION

EXPLANATION

CONSTRUCTION OF WHOLE SYSTEMS

"KNOWLEDGE ENGINEERING"





This shows the particular representation which we have chosen.

We put the fact triple in the inference rules. If you have a certain number of facts, then you can come to certain conclusions, often only weakly suggested.

FACT TRIPLE

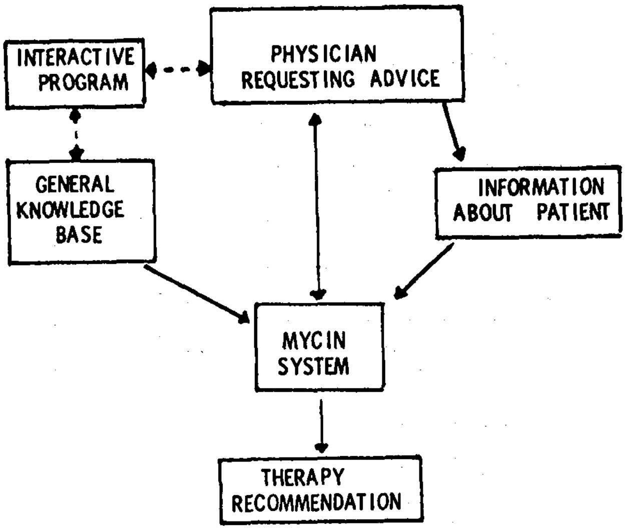
THE $\underline{X}$ OF $\underline{Y}$ IS $\underline{Z}$ (WITH CF CERTAINTY)

RULE

IF FACT_1 & FACT_2 & ... & FACT_N

THEN CONCLUDE FACT_0 (WITH CERTAINTY CF)

MYCIN SYSTEM
FOR RECOMMENDING THERAPY FOR INFECTIOUS DISEASES



This is an example of how MYCIN works. This is used for research only. We used this for testing whether you can use this very simple representation of knowledge to build a program capable of expert performance. The system is asking questions of the user and based on the answers it reaches conclusions.

USER'S DESCRIPTION OF PATIENT

```
This will be patient 538
If you would care to enter a brief summary of PATIENT-538's Case,
to be used for easy future reference to this file, please do so now
(type as many lines as required, terminate with a control-Z):

** 34 yr old white male with severe burn. After 15 days
** of hospitalization pt became disoriented and febrile. On 16th
** hospital day LP performed. PMN 100, PMN %99, Pro 280, Glu 5
** (periph blood glu unk). WBC 1900 with 88% PMN and 5% bands.
** Pt had drug-induced aplastic anemia. Current med at time of LP:
** Cephalothin. Final org: Pseudomoas-aeruginosa. Correct Rx: Ideal
** would have been Carb + Tobra or (Genta). Same as pt 504 except
** gram stain of CSF omitted.
**   Z
```

The system asks questions like these.

```
INFECTIOUS DISEASE CONSULTATION SESSION
        (USER'S RESPONSES FOLLOW DOUBLE ASTERISKS)

-------------------PATIENT-538------------------

1)  Patient's name:
**  PT538

2)  Age:
**  34 YEARS

3)  Sex:
**  MAEL
= MALE

4)  Race:
**  WHY
```

The system can be asked questions about why some information is required as shown here.

(i.e. WHY is it important to determine the race of PATIENT -538?)

(1.0) The race of a patient is one of the 5 standard parameters that identify a patient.

This may be useful later in the consultation. For example, it is relevant in determining the organisms (other than those seen on cultures or smears) which might be causing the infection.

(back to question 4 ...)

** CAUCASIAN

We skipped some questions to (14)

Here we have also an intermediate conclusion at the bottom.

14) Has Pt538 recently had symptoms of persistent headache or other abnormal neurologic symptoms (dizziness, lethargy, etc.)?

** YES

15) Has Pt538 recently had objective evidence of abnormal neurologic signs (nuchal rigidity, coma, seizures, etc.) documented by physician observation or examination?

** YES

The CSF cultures will be considered to be associated with MENINGITIS.

It also allows entering blocks of data as shown here and on the following few slides.

Please enter the results of pending cultures in the following table.

	site	culture number	date collected	stain examined	method of collection (urine, sputum only)
6) **	CSF	1234	30-Jan-77	YES	*

23) For how many days has Pt538 had abnormal neurologic signs?

** 7.2 HOURS

24) For how many days has Pt538 had neurologic symptoms?

** 1 DAYS

25) Has Pt538 had recent evidence of tender or enlarged salivary glands?

** NO

26) Has Pt538 been exposed to any contagious disease recently (e.g. meningococcal disease, mumps)?

** NO

Please enter CSF findings in the following table

	CSF WBC	Percent PMN	Protein	CSF Glucose	Peripheral Glucose
27)**	100	99	280	5	UNKNOWN

38) Has pt538 ever undergone splenectomy?

** NO

39) Is Pt538 a burn patient?

** WHY

(i.e. WHY is it important to determine whether PATIENT-538 has been seriously burned?)

(4.0) This will aid in determining the organisms (other than those seen on cultures or smears) which might be causing the infection.
It has already been established that

(4.1) the infection which requires therapy is meningitis, and

(4.2) organisms were not seen on the stain of this pending csf culture,

and

(4.3) the type of the infection is bacterial

Therefore, if

(4.4) pt538 has been seriously burned

then

there is suggestive evidence (.5) that pseudomonas-aeruginosa is one of the organisms (other than those seen on cultures or smears) which might be causing the infection.

Author: Yu
(RULE578)

Going on like this for about 40-80 questions and answers we come to another set of intermediate conclusions with sufficient evidence.

My therapy recommendations will be designed to treat for organisms that either are very likely or, although less likely, would have a significant effect on therapy selection if they were present. It is important to cover for the following probable infection(s) and associated organism(s):

INFECTION-1 is MENINGITIS

+ (ITEM-1) E. COLI (from clinical evidence only)
+ (ITEM-2) PSEUDOMONAS-AERUGINOSA (from clinical evidence only)
+ (ITEM-3) KLEBSIELLA-PNEUMONIAE (from clinical evidence only)
+ (ITEM-4) DIPLOCOCCUS-PNEUMONIAE (from clinical evidence only)

[Considering AMPICILLIN-AND-GENTAMICIN for use against INFECTION-1 ...]

42) Do you have reason to suspect that Pt538 may have impaired renal function?

** NO

43) Pt538's weight in kilograms (or number POUNDS):

** 70

Finally we come to the recommendation by the system given here.

REC-1 My preferred therapy recommendation is as follows:

In order to cover for Items <1 2 3 4>:
Give the following in combination:

1) AMPICILLIN
Dose: 3.5 g (28.0 ml) q4h IV (calculated on basis of
50 mg/ kg)

2) GENTAMICIN
Dose: 119 mg (3.0 ml, 80 mg / 2 ml ampule)
q8h IV (calculated on basis of
1.7 mg/kg) plus consider giving 5mg q24h Intrathecal

Comments: Monitor serum concentrations

Since high concentrations of penicillins can inactivate aminoglycosides, do not mix these two antibiotics in the same IV bottle.

If for some reason we can not implement the best recommendations, we may ask for the next best recommendations shown here.

[REC-2] Next best therapy recommendation :

In order to cover Items <1 2 3>:
Give: GENTAMICIN
Dose: 119 mg (3.0 ml, 80mg/2ml ampule) q8H IV
(calculated on basis of 1.7 mg/kg) plus consider
giving 5mg q24h Intrathecal

Comments: Monitor serum concentrations

In order to cover for Item <4>:
Give: PENICILLIN
Dose: 3.5 million units (7.0 ml) q4 h IV (calculated
on basis of 50000 units/kg)

Here is how well the system did in a trial.

The experts were asked blindly. They did not know that the results would be compared to computer results. MYCIN was performing as well as the faculty at Stanford.

Ratings by 8 Experts on 10 Meningitis Cases

(Perfect Score = 80) *

MYCIN	52	Actual Therapy	46
Faculty-1	50	Faculty-4	44
Faculty-2	48	Resident	36
Inf. Dis. Fellow	48	Faculty-5	34
Faculty-3	46	Student	24

* Equivalent Therapy or Acceptable Alternate = 1

Unacceptable Therapy = 0

Here is a simple rule with extra information which may further justify the final recommendation. For instance, where does Rule 300 come from?

RULE300

If: 1) The infection which requires therapy is meningitis, and
2) Sickle-cell-disease is one of the relevant items from the history of the patient

Then: There is weakly suggestive evidence (.35) that diplococcus-pneumoniae is one of the organisms for which therapy should cover

AUTHOR: Yu

LITERATURE:

G. Karalazin "Sickle-cell anemia - clinical manifestations in 100 patients" AmJMedSci 269:51 1975.
E. Barret-Connor "Acute pulmonary disease and sickle-cell anemia" ARRD 104:159 Aug 1971
M. Robinson "Pneumococcal meningitis in sickle-cell anemia "NEJM 274:1006 1966

JUSTIFICATION: None

PATIENTS: None

ORIGINAL DATE: 24-NOV-76 12:37:29
LAST EDIT DATE: 24-NOV-76 12:37:29

RULE FORMAT

```
IF:  1) THE STAIN OF THE ORGANISM IS
        GRAMNEG, AND
     2) THE MORPHOLOGY OF THE ORGANISM IS
        ROD, AND
     3) THE PATIENT IS A COMPROMISED HOST
THEN:   THERE IS SUGGESTIVE EVIDENCE (.6) THAT
        THE IDENTITY OF THE ORGANISM IS
        PSEUDOMONAS
```

```
($AND (SAME CNTXT GRAM GRAMNEG)
      (SAME CNTXT MORPH ROD)
      (SAME CNTXT COMPROMISED))

(CONCLUDE CNTXT IDENT PSEUDOMONAS TALLY 600)
```

RULE 035

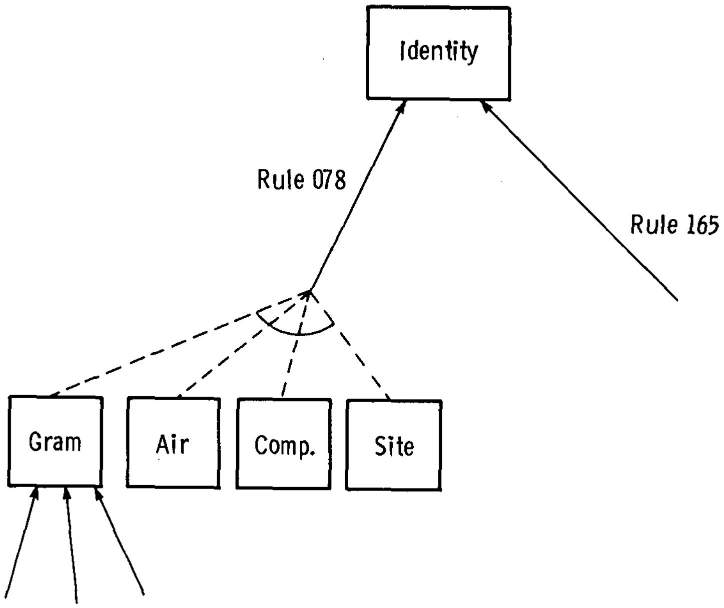


Illustration of MYCIN and/or Subgoal
Tree with Depth First Search

It is also important to answer questions which the system did not do and why.

** WHY DIDN'T YOU GIVE TETRACYCLINE FOR E. COLI IN REC-1?

Shall I describe the therapy algorithm for you? ** NO

TETRACYCLINE was discounted for ITEM-1 because there is evidence that this e. coli is not sensitive to it.

40) Has Herbert Levin been exposed recently to MUMPS?

** N

41) Does Herbert Levin have a history of SICKLE-CELL-DISEASE?

** COMMENT

Comment on subject: Redundant question

Please type your comments below, using as many lines as required, terminating your message with a blank line.

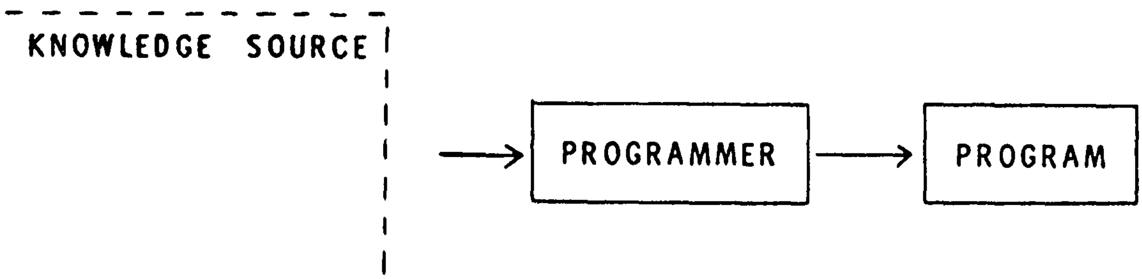
**Pt 504 Mycin asked if he had "sickle-cell-disease" and the patient was known to be white (caucasian)

**

Thank you for your comments We will respond soon. You may now continue from the point at which you typed "COMMENT".

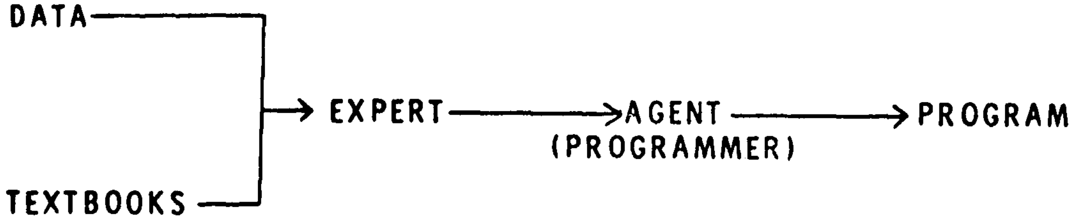
** NO

This brings us to the problem of transferring the knowledge. The programmer cannot read a textbook and put it into the program. He needs to be knowledgeable in the subject area, he needs to be an expert to formulate the right statements and questions.



KNOWLEDGE TRANSFER

The expert needs to do the translation for the programmer as shown in the next slide. We tried to bypass the expert and programmer by writing general induction programs forming general rules.



GIVING KNOWLEDGE TO A PROGRAM

("KNOWLEDGE ACQUISITION")

This program was written a few years ago. Here the expert has seen a recommendation, but he has expected something else. The system now tries to find out what the expert knows, which it does not know, and it queries the expert as shown in the next slides.

My therapy recommendation will be based on the following probable infection(s) and potential causative organism(s):

INFECTION-1 is PRIMARY-BACTEREMIA

<Item 1>E.COLI [ORGANISM-1]

<Item 2>PSEUDOMONAS-AERUGINOSA [ORGANISM-1]

Are all these values correct?

****** N**

Are there any values missing?

****** Y**

Which one(s)?

****** KLEBSIELLA-PNEUMONIAE PROTEUS-NON-MIRABILIS**

Are there any values that shouldn't have appeared?

****** Y**

Which one(s)?

***** PSEUDOMONAS-AERUGINOSA**

Is this rule correct?
++**Y
Should its premise have failed for this case?
++**Y
Which clause of the premise should have been false? [give.]
++**1

No conclusions were reached concerning the category of ORGANISM-1.

Should it have been possible to conclude a value?
++**Y

We'll start with the PSEUDOMONAS-AERUGINOSA first.

The following rules were used in deducing that the identity of ORGANISM-1 is pseudomonas-aeruginosa

RULE 184

Since 1) the category of ORGANISM-1 is not known
2) the gram stain of ORGANISM-1 is gramneg
3) the morphology of ORGANISM-1 is rod
4) the aerobicity of ORGANISM-1 is facul

There is weakly suggestive evidence (.3) that the identity of ORGANISM-1 is pseudomonas-aeruginosa

Which one?
++** ENTEROBACTERIACEAE

There were no rules successfully invoked in the attempt to deduce that the category of ORGANISM-1 is enterobacteriaceae

The following (potentially applicable) rules were NOT executed because the **/d clause failed.

(RULE 116 AND RULE 050 ARE PRINTED OUT FOR INSPECTION)

Are all these rules correct?
****Y
Should any of them have been successfully invoked?
****N

Is there a rule missing?
****Y

Ok, then I need a rule that would allow me to deduce that the category or ORGANISM-1 is enterobacteriaceae

Would you like to give me such a rule now?
****Y

Good...

This is my understanding of your rule:
RULE 383

If: 1) The infection is primary-bacteremia, and
2) The site of the culture is one of:
those sites that are normally sterile
Then: There is strongly suggestive evidence (.8) that the
category of the organism is enterobacteriaceae
Okay? (Yes or No)
—**Y

The system now tries to remind the expert of things he might have forgotten. There is here some intelligence involved.

```
I hate to criticize, Dr. Davis, but did you know that most rules
about what the category of an organism might be, that mention-
          the infection
          the site of a culture
ALSO mention-
          [A] - the portal of entry of an organism
Shall I try to write a clause to account for [A]?
++**Y
How about-
          [A] The portal of entry of the organism is gi
Ok?
++**Y
```

A big problem is working on checking rules. Some of the checks are indicated here.

CHECKS ON RULES

LOGICAL INCONSISTENCY WITH EXISTING RULES

OVERLAP WITH EXISTING RULES

COMPLETENESS OF RULE

SUCCESS FOR CORRECTING PERCEIVED ERRORS

EFFECTS ON TEST CASES IN LIBRARY

In this case, the system has meta rules, higher level rules, or strategy rules; in this case, medical strategies.

METARULE 002

If 1) the patient is a compromised host, and
2) there are rules which mention in their premise pesudomonas
3) there are rules which mention in their premise klebsiellas

There is suggestive evidence (.4) that the former should be done before the latter.

Everything mentioned was in the context of MYCIN. We are in fact working with several different programs, and I'd like to summarize the state of the art here.

MYCIN has about 450 rules, on the order of 1000 to 2000 auxiliary facts about medicine, drugs, etc. It has a vocabulary of about 2000 words, but the domain is very narrow.

We feel very constrained that we need one expert as "knowledge Czar." We do not know enough to automate the input from various experts and integrate it.

EXPERT SYSTEMS: STATE OF THE ART

- NARROW DOMAIN OF EXPERTISE
- LIMITED LANGUAGE FOR EXPRESSING FACTS AND RELATIONS
- LIMITING ASSUMPTIONS ABOUT PROBLEM AND SOLUTION METHODS (HELP REQUIRED FROM A "KNOWLEDGE ENGINEER")
- STYLIZED I/O LANGUAGES
- KNOWLEDGE BASE EXTENSIBLE ABOVE A THRESHOLD
- LITTLE KNOWLEDGE OF OWN SCOPE AND LIMITATIONS
- STYLIZED EXPLANATIONS OF LINE OF REASONING
- SINGLE EXPERT AS "KNOWLEDGE CZAR"

PROBLEM SOLVING IN A DISTRIBUTED ENVIRONMENT

Richard F. Rashid
Carnegie-Mellon University

COMPONENTS

DISTRIBUTED PROBLEM SOLVING IS A BLEND OF TWO DISCIPLINES:

1. PROBLEM SOLVING AND AI

2. DISTRIBUTED SYSTEMS (MONITORING)

Examples of a multi-process language environment include MESA, SMALL-TALK, SAIL, SIMULA, and ADA. (See, for example, Mitchell, 1978).

Multi-programmed single processors represent the most common class of computer systems. Examples include the POP-10 running TOPS-10, TOPS-20, or TENEX, the VAX running UNIX, etc.

Carnegie-Mellon's CM-* is a good example of a tightly-coupled multiprocessor. It consists of 55 LSI-11's connected by a multi-level bus structure. [Srwiorek, 1978].

Xerox Alto's or Three-River's Corporation Perqs connected via an ethernet are good examples of loosely-coupled processors. The distinction between tightly-coupled and loosely-coupled is made primarily on the basis of physical distance and, therefore, delay in communication. Bandwidth can remain quite high in loosely-connected systems. For example, two Xeros Alto's connected via an ethernet actually can communicate at higher bandwidth than two distant LSI-11's Cray-1.

WHAT IS IT?

PROBLEM SOLVING WITH MULTIPLE LOGICAL PROCESSING ELEMENTS

- MULTIPLE-PROCESS LANGUAGE ENVIRONMENTS
- MULTI-PROGRAMMED SINGLE PROCESSORS
- TIGHTLY-COUPLED MULTI-PROCESSORS
- LOOSELY COUPLED PROCESSORS (NETWORKS)

It may be necessary to distribute because the application itself is one of managing distributed resources (e.g., distributed sensor net) and communication delays preclude centralized processing.

It may be desirable to distribute because a single computational engine (even a Cray-1) may not satisfy the needs of a given task.

In addition, considerations of reliability may dictate distribution.

See subsequent slides.

REASONS:

- HAVE TO DISTRIBUTE
- NEED TO DISTRIBUTE
- WANT TO DISTRIBUTE

HAVE TO DISTRIBUTE:

- PROBLEM ITSELF MAY BE DISTRIBUTED
  - DISTRIBUTED SENSORS, HIGH COMMUNICATION COSTS
- SPECIALIZED HARDWARE REQUIRED
- PROBLEM MAY BE LARGER THAN CAN BE HANDLED BY A SINGLE PROCESSOR

NEED TO DISTRIBUTE:

- PROBLEM MAY BE TOO COMPLEX

WANT TO DISTRIBUTE:

- ALGORITHM MAY BE INHERENTLY PARALLEL

- REDUNDANCY AND RELIABILITY

- CHEAPER OR FASTER TO USE A NUMBER OF SMALL PROCESSORS RATHER THAN ONE LARGE PROCESSOR

  - HARPY MACHINE: 4 LSI-11's FASTER THAN 1 KL-10 FOR SPECIAL TASK

The HARPY machine built at CMU is a good example of a simple, cheap multiprocessor which takes advantage of parallelism to outperform a much faster uniprocessor. The HARPY machine consists of five LSI-11's connected by a bus to a "smart" memory - a bit-slice microprocessor which implements in microcode the access to the HARPY database. The memory box-microprocessor is well matched in speed to the needs of the LSI-11's so that maximal parallelism is achieved.

TRADITIONAL PROBLEM SOLVING

- ACQUIRE KNOWLEDGE (USUALLY FROM A THEORY, OR FROM EXPERTS, OR FROM PROGRAMMER)

- GATHER RAW MATERIALS (DATA FROM OUTSIDE ENVIRONMENT)

- SEARCH FOR SOLUTION

If the search space is small, exhaustive search is almost always the best approach. A large search space, however, may make exhaustive search prohibitively expensive even for the fastest computers. Thus, some decisions must be made as to what part of the solution space to search.

SEARCH

SMALL SPACE => NO PROBLEM

LARGE SPACE => DECISION MAKING (HEURISTICS)

Branch and bound is a classical algorithm inherited from the field of operations research. The BEAM [Loweure, 1976] search was developed at Carnegie-Mellon University as part of the HARPY system. MSYS is yet another search strategy developed by Marty Tenenbaum and Harry Barrow at SRI. NOAH is a planning system developed by Sacerdoti, also while he was at SRI [Sacerdoti, 1977]. Decision theory is a statistical theory utilized by a number of AI systems such as Yakimovsky's image segmentation program done at Stanford in 1974 [Feldman, 1974].

SEARCH TECHNIQUES

EXAMPLES OF TECHNIQUES USED TO HELP MAKE DECISIONS ABOUT THE EXPLORATION OF A SEARCH SPACE:

- BRANCH AND BOUND
- BEAM
- MSYS
- PLANNING (NOAH)
- DECISION THEORY

COMPLEXITY vs. UNCERTAINTY

TWO BASIC PROBLEMS OCCUR AGAIN AND AGAIN IN THE DESIGN AND IMPLEMENTATION OF PROBLEM SOLVING SYSTEMS:

- COMPLEXITY
- UNCERTAINTY

COMPLEXITY

COMPLEXITY IS A CLASSIC PROBLEM NOT ONLY IN AI BUT ALSO IN SYSTEMS. IT AFFECTS THE WAY IN WHICH A PROGRAMMER WILL SOLVE A PROBLEM. IT CAUSES THE PROGRAMMER TO:

- PARTITION A PROBLEM INTO TRACTABLE COMPONENTS
- AND MINIMIZE INTERACTION BETWEEN COMPONENTS

BOTH IN PROBLEM SOLVING AND OPERATING SYSTEMS INCREASED COMPLEXITY HAS LED TO HIERARCHICAL STRUCTURING OF PROGRAMS, MODULARIZATION, AND "DISTRIBUTED PROCESSING" ON SINGLE PROCESSORS.

UNCERTAINTY

AS UNCERTAINTY INCREASES, GREATER EMPHASIS IS PLACED ON:

- COOPERATION BETWEEN COMPONENTS
- LARGE, SHARED KNOWLEDGE BASES
- THE HANDLING OF COMPETING HYPOTHESES
- TECHNIQUES FOR RESOLVING CONFLICTS DUE TO DECISIONS MADE WITH UNCERTAIN DATA

HUMAN ORGANIZATIONS

HUMAN ORGANIZATIONS ARE DISTRIBUTED PROBLEM SOLVING SYSTEMS (FOX, 1979). THEY CONSIST OF:

- COLLECTIONS OF INDEPENDENT PROCESSORS, EACH WITH A LARGE DATA BASE OF KNOWLEDGE AND SOPHISTICATED DEVICES FOR SENSORY INPUT
- NARROW BANDWIDTH COMMUNICATIONS CHANNELS BETWEEN PROCESSORS (SLOW SPEED, LONG DELAYS)

BOUNDED RATIONALITY

A KEY CONCEPT IN THE THEORY OF HUMAN ORGANIZATIONS IS "BOUNDED RATIONALITY":

- CONTROL BOUNDS
- INTERPRETATION BOUNDS
- DECISION BOUNDS

SAMPLE ORGANIZATIONS

SINGLE HUMAN

SMALL GROUP

SIMPLE HIERARCHY

COMPLEXITY

UNCERTAINTY

UNIFORM HIERARCHY

HETERARCHY

MARKET

"Abstraction" means collecting data, analyzing it, and condensing it into a more meaningful form. For example, a speech processing unit may abstract a set of sounds into the form of phonemes or even words. Abstraction reduces the volume of communication and, therefore, its costs.

"Omission" may be possible in some cases. Certain information collected at various levels of the system can be recognized as unnecessary and simply forgotten.

A problem may be decomposed into sub-problems which are independent and do not require communication.

DECOMPOSITION

IN DECOMPOSING A SYSTEM, THE GOAL IS TO ACHIEVE INTERACTION LOCALITY. SYSTEMS WHICH EXHIBIT INTERACTION LOCALITY ARE CALLED "NEARLY DECOMPOSABLE." TWO DIVISION TECHNIQUES USED BY CORPORATIONS ARE:

1. PRODUCT DIVISION
2. FUNCTIONAL DIVISION

REDUCING COMMUNICATION COSTS

COMMUNICATION COSTS CAN BE REDUCED BY:

- ABSTRACTION
- OMISSION
- DECOMPOSITION

HANDLING UNCERTAINTY

IN HUMAN ORGANIZATIONS, UNCERTAINTY CAN BE HANDLED IN A NUMBER OF WAYS (GALBRAITH, 1973):

- SLACK RESOURCES
- SELF-CONTAINMENT
- VERTICAL INFORMATION PROCESSING
- LATERAL RELATIONS

STRUCTURES FOR
DISTRIBUTED PROBLEM SOLVING

LESSER AND CORKILL (LESSER, 1979) HAVE PROPOSED DIFFERENTIATING DISTRIBUTED PROBLEM SOLVING SYSTEMS INTO TWO CLASSES:

- COMPLETELY ACCURATE/NEARLY AUTONOMOUS
- FUNCTIONALLY ACCURATE/COOPERATIVE

EXAMPLE SYSTEMS

SEVERAL DISTRIBUTED PROBLEM SOLVING SYSTEMS HAVE ALREADY BEEN CONSTRUCTED OR PROPOSED:

- HEARSAY II (HERMAN, LESSER, REDDY, ET AL) [LESSER, 1979]
- DISTRIBUTED HEARSAY ARCHITECTURE (ERMAN, LESSER)
- HARPY MACHINE (BISIANI, MAUERSBERN, REDDY)
- DISTRIBUTED NOAH (CORKILL) (SEE [LESSER, 1979])
- TRAFFIC CONTROL (DISTRIBUTED RELAXATION) (BROOKS, LESSER)
- CONTRACT NETS (SMITH) [SMITH, 1977]

HEARSAY II

HEARSAY II WAS A SYSTEM DESIGNED TO UNDERSTAND HUMAN UTTERANCES.

- LARGE DEGREE OF UNCERTAINTY
- STRONGLY HIERARCHICAL
  - -HIERARCHICAL BLACKBOARD (PHONETIC, SYNTACTIC, AND SEMANTIC LEVELS)
  - -HIERARCHICAL OF CONTROL (KS's)
- CENTRALIZED DATA BASE AND SCHEDULER

DISTRIBUTED HEARSAY II

HEARSAY II ARCHITECTURE WAS MODIFIED TO HANDLE A PROBLEM USING DISTRIBUTED SENSORS.

- ARCHITECTURE:
  - -NUMBER OF NODES EACH WITH ITS OWN KS's AND BLACKBOARDS (HETERARCHY OF HIERARCHIES)
  - -HYPOTHESES EXCHANGED AMONG ALL NODES (ONLY "BEST" EXCHANGED)
  - -COMMUNICATION UNCERTAINTY CONSIDERED PART OF OVERALL PROBLEM UNCERTAINTY

DISTRIBUTED HEARSAY II

- RESULTS:

  - SYSTEM PERFORMED EVEN WITH UP TO 50% OF MESSAGES LOST.

  - LOSS OF SOME MESSAGES ACTUALLY IMPROVED PERFORMANCE.

HARPY MACHINE

- THE HARPY SPEECH UNDERSTANDING ALGORITHM WAS USED TO DESIGN A SPECIFIC MULTI-PROCESSOR.

- HARPY MACHINE: 5 LSI-11's AND SPECIAL MICROPROGRAM DATA BASE PROCESSOR

- HARPY MACHINE IS AN EXAMPLE OF DISTRIBUTED PROBLEM SOLVING WITH A CENTRALIZED DATA BASE. DATA BASE WAS ACTUALLY FASTER THAN PROCESSORS AND THE SYSTEM AS A WHOLE PERFORMED HARPY ALGORITHMS FASTER THAN ANY PREVIOUSLY USED MACHINE, EVEN KL-10.

DISTRIBUTED TRAFFIC CONTROL

- RELAXATION ALGORITHM ADAPTED TO DISTRIBUTED TRAFFIC CONTROL SYSTEM.
- RESULTS: POOR (TRASHING, PERFORMANCE DEGRADATION)
- REASONS: TOO LITTLE "RATIONALITY" AT NODE

CONTRACT NETS

- A CONTRACT NET IS THE COMPUTER REALIZATION OF A MARKET STRUCTURE IN HUMAN ORGANIZATION.
- PROCESSES MAKE CONTRACTS AND SUPPLY SERVICES.
- BASIC DECISION MAKING DEVICE IS "PRICE."
- AS WITH HUMAN ORGANIZATIONS, CONTRACT NETS AID IN REDUCING COMPLEXITY BUT ARE UNABLE TO HANDLE UNCERTAINTY.

DISTRIBUTED SEARCH

DISTRIBUTED PROBLEM SOLVING IMPLIES DISTRIBUTED SEARCH. DIFFERENT APPROACHES EXIST IN SPECIFYING THE NATURE OF THE DATA BASE BEING SEARCHED:

- SYNTHETIC DATA BASE -- EACH NODE HAS A LOCAL DATA BASE WHICH MAY NOT BE CONSISTENT WITH THAT OF OTHER NODES. THE DATA BASE AS A WHOLE IS THE SYNTHESIS OF THE DATA BASES OF ALL NODES TAKING INTO ACCOUNT CONFLICTS.

- DECOMPOSED DATA BASE -- EACH NODE HAS A GLOBALLY CONSISTENT PORTION OF THE WHOLE DATA BASE. WHOLE IS JUST THE SUM OF ITS PARTS.

SYSTEM ISSUES IN DISTRIBUTED PROBLEM SOLVING

[RASHID, 1980]

ONE REASON THAT FEW TRULY DISTRIBUTED PROBLEM SOLVING SYSTEMS EXIST HAS BEEN THE LACK OF TOOLS.

SOME OF THE ISSUES IN DESIGNING TOOLS FOR DISTRIBUTED PROBLEM SOLVING ARE:

- LOGICAL ROLE OF COMMUNICATION
- BINDING AND ADDRESSING
- HANDLING OF ASYNCHRONOUS EVENTS, ERRORS
- FAULT TOLERANCE

CMU IPC

AN INTER-PROCESS COMMUNICATION FACILITY IS BEING BUILT AT CMU FOR DISTRIBUTED PROBLEM SOLVING.

MAIN FEATURES:

- PROVIDES LOGICAL CONCEPT OF COMMUNICATION INDEPENDENT OF OS OR NETWORKING
- COMPLETLY ASYNCHRONOUS (MESSAGE-BASED)
- LOCALLY DEFINED BUT HAS NATURAL AND TRANSPARENT NETWORK EXTENSION

CMU IPC

MAIN FEATURES:

- COMMUNICATION RATHER THAN PROCESS ORIENTED
- ABILITY TO HANDLE BOTH CA/NA AND FA/C ALGORITHMS
- ABILITY TO EXPLICITLY MANAGE COMMUNICATION LINKS

PORTS

A PORT IS:

- A KERNEL OBJECT
- A QUEUE FOR MESSAGES
- KNOWN TO PROCESSES ONLY THROUGH LOCAL NAME

USE OF PORTS

- IDENTIFY TRANSACTION
- IDENTIFY DATA STRUCTURE (IN ANOTHER PROCESS)
- ISOLATE COMMUNICATION

MESSAGES

- A MESSAGE IS A COLLECTION OF TYPED DATA OBJECTS.
- A MESSAGE IS SENT FROM A PROCESS TO A PORT.
- MESSAGES ARE HANDLED BY THE SYSTEM ACCORDING TO THEIR TYPE (I.E., SERVICE CLASS).
- NAMES, ACCESS AND OWNERSHIP TO PORTS CAN BE PASSED IN MESSAGES.

MESSAGE TYPES

EACH MESSAGE HAS A MESSAGE TYPE WHICH DETERMINES ITS SERVICE CLASS. A MESSAGE'S TYPE DETERMINES ITS:

- RELIABILITY
- SEQUENTIALITY
- MAXIMUM AGE
- PRIORITY
- SENSITIVITY TO FLOW CONTROL

MESSAGE TYPES ARE CONSIDERED ADVISORY

PORT ACCESS LIST

- PORTS ARE KNOWN TO PROCESSES ONLY AS LOCAL NAMES
- EACH PROCESS HAS IN THE KERNEL AN ACCESS LIST OF PORTS
- PROCESSES MAY ONLY REFER TO PORTS BY THEIR LOCAL NAMES. (NO ACCIDENTAL ACCESS.)
- KERNEL HAS COMPLETE KNOWLEDGE OF COMMUNICATION.

FLOW CONTROL ALTERNATIVES

POSSIBLE FLOW CONTROL ALTERNATIVES:

1. INFINITE MESSAGE QUEUES
2. EXPLICIT FLOW CONTROL MESSAGES
3. AUTOMATIC QUEUE BACKPRESSURE
    - BASED ON SENDER'S OUTSTANDING MESSAGES
    - BASED ON MESSAGES WAITING FOR PORT

FLOW CONTROL IN CMU IPC

NORMAL CASE:

- IF A PROCESS ATTEMPTS TO SEND A MESSAGE TO A FULL PORT OF ANOTHER PROCESS, IT WILL BE SUSPENDED UNTIL THE REQUEST CAN BE COMPLETED.

FLOW CONTROL IN CMU IPC

EXCEPTIONS:

- A PROCESS CAN ASK TO BE IMMEDIATELY NOTIFIED RATHER THAN SUSPENDED IF THE RECEIVING PORT IS FULL; OR

- A PROCESS CAN REQUEST TO RECEIVE NOTIFICATION WHEN THE MESSAGE CAN BE ACCEPTED (IN THE FORM OF A MESSAGE FROM THE KERNEL).

EXCEPTION HANDLING

- PROCESSES MAY ASSOCIATE SOFTWARE INTERRUPT HANDLER WITH MESSAGE RECEPTION
- ASYNCHRONOUS ERRORS ARE SIGNALLED ACROSS PROCESS BOUNDARIES WITH EMERGENCY MESSAGES
- SINCE KERNEL KNOWS WHICH PROCESSES ARE COMMUNICATING, IT CAN NOTIFY PROCESSES OF PARTNER'S DEATH

EMERGENCY MESSAGES

- EMERGENCY MESSAGES HAVE HIGHEST PRIORITY, ARE SPECIALLY FLOW CONTROLLED, AND HAVE NO MAXIMUM AGE.
- USED TO ALLOW URGENT INFORMATION TO BE FORWARDED BETWEEN PROCESSES INDEPENDENT OF MESSAGE BACKLOG.
- PURPOSE:
    - ERROR NOTIFICATION
    - SPECIAL EVENT PROCESSING
    - DEBUGGING

ACQUIRING A NAME

A PROCESS 'A' MAY ACQUIRE THE NAME OF ANOTHER PROCESS 'B' IF:

- 'B' IS THE FATHER OF 'A'
- 'B' SENDS 'A' A MESSAGE
- 'A' KNOWS THE STRING NAME FOR 'B' AND REQUESTS 'B' NAME FROM THE NAME SERVER
- A PROCESS 'C' WHICH KNOWS 'B' SENDS A MESSAGE TO 'A' AND TELLS IT ABOUT 'B'

NAME SEARCH

A DYNAMIC CATALOGUE OF STRING NAMES AND ASSOCIATED PROCESS-PORT NUMBERS

- PROVIDES LINK BETWEEN UNRELATED PROCESSES WHICH WISH TO COMMUNICATE
- PROVIDES ACCESS TO OTHER SERVICES (E.G., FILE SYSTEM, ARPANET, ETC.)
- ALLOWS FOR CREATION OF GENERIC PROCESSES ON REQUEST
- LOCATES PROCESSES AND SERVICES ON FOREIGN HOSTS (THROUGH NETWORK SERVERS)

NETWORK SERVERS

- EXIST AT PROCESS LEVEL (NOT KERNEL LEVEL)
- SERVE AS 'SHADOWS' FOR EXTERNAL PROCESSES
- CONVERT MESSAGE FORMATS, DATA TYPES, ETC.
- SERVICE COMMUNICATION LINKS AND PROTOCOLS
- PROVIDE EXTERNAL NAME SERVICE (OPTIONAL)

ADVANTAGES

- CAPABILITY-BASED COMMUNICATION WHICH ALLOWS TRANSPARENT REDIRECTION OF MESSAGES
- AUTOMATIC NOTIFICATION OF PROCESS DEATH
- EMERGENCY MESSAGES FOR ERROR HANDLING
- REDIRECTION OF NAME BINDING REQUESTS

IMPLEMENTATION:

- VAX/UNIX (MARCH, 1980)

PLANNED USE:

- CMU'S TESTBED DISTRIBUTED SENSOR BED

REFERENCES

Feldman, J.A., and Y. Yakimousky, (1974) Decision Theory and Artificial Intelligence: A Semantic Based Region Analyzer, Artificial Intelligence, Vol. 5, pp. 349-371.

Fox, Mark, (1979), Organization Structures: Structuring Complex Program, Technical Report, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

Galbraith, Jay, (1973), Designing Complex Organizations, Addison-Wesley.

Lesser, V., (1979) Working Papers in Distributed Computation. I - Cooperative Distributed Problem Solving, Distributed Computation Group, Computer and Information Science, University of Massachussets.

Lowerre, B. (1976), The HARPY Speech Recognization System (Ph.D. Thesis), Technical Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

Mitchell, J. A., W. Maybury, and R. Sweet, (1978), "Mesa Language Manual," Technical Report, Xerox Parc, February, 1978.

Rashid, R., (1980), An Inter-Process Communication Facility for Universities, Technical Report, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

Sacerdoti, E. D. (1977), A Structure for Plans and Behavior, American Elsevier, New York.

Siewiorek, O., V. Kini, H. Mashburn, S. McConnel, and M. Tsao, (1978), A Case Study of C.ump, CM*, and C.ump: Part I - Experiences with Fault Tolerance in Multiprocessor Systems, Proceedings of the IEEE, Vol. 66, No. 11, November, 1978.

Smith, R. A., (1977), The Contract Net: A Formalism for the Control of Distributed Problem Solving, Fifth International Joint Conference on Artificial Intelligence, Cambridge, Masschussets.

HUMAN PERFORMANCE CHARACTERISTICS IN RELATION TO AUTOMATED DECISION MAKING AND PROBLEM SOLVING

WILLIAM R. FERRELL
UNIVERSITY OF ARIZONA

HUMAN PERFORMANCE CHARACTERISTICS IN RELATION TO AUTOMATED DECISION MAKING AND PROBLEM SOLVING

WILLIAM R. FERRELL
SYSTEMS AND INDUSTRIAL ENGINEERING DEPARTMENT
THE UNIVERSITY OF ARIZONA
TUCSON, AZ 85721

1. INTRODUCTION

HUMAN CHARACTERISTICS AND ESPECIALLY HUMAN LIMITATIONS SHOULD PROVIDE INPUT FOR THE DESIGN OF AUTOMATED PROBLEM SOLVING AND DECISION MAKING SYSTEMS, AND IN SOME CASES SHOULD BE THE GOVERNING FACTORS. THERE ARE DIFFERENT REASONS FOR THIS DEPENDING ON THE TYPE OF SYSTEM INVOLVED.

1. REMOTE OR UNTENDED SYSTEMS ARE EXPECTED TO PERFORM, IN SOME SENSE, AS WOULD A PERSON WERE HE THERE, SO THE DEFAULT CONDITIONS MUST BE ONES THAT SUCH A PERSON WOULD JUDGE AS REQUIRING HIS INTERVENTION. THE FUNCTIONING OF SYSTEMS THAT ARE REMOTE AND RELATIVELY AUTONOMOUS SHOULD EXCEED, IF POSSIBLE, THE CAPABILITIES OF A PERSON. BUT THERE WILL NECESSARILY BE BOUNDS TO THIS COMPETENCE. RECOGNITION OF THEM AND RAPID AND ACCURATE TRANSFER OF CONTROL RESPONSIBILITY, STATUS INFORMATION, AND DIAGNOSTIC INFORMATION REQUIRES CAREFULLY DESIGNED INTERFACING WITH HUMAN CHARACTERISTICS. THERE IS ALWAYS A MAN-MACHINE INTERFACE!

2. DECISION AND PROBLEM SOLVING SYSTEMS THAT ARE EXPECTED TO AUGMENT HUMAN CAPABILITIES THROUGH A COOPERATIVE INTERACTION PLACE A PREMIUM ON THERE BEING AN EASY AND NATURAL COMMUNICATION WITH THE OPERATOR/SUPERVISOR. FOR EFFICIENCY, THESE SYSTEMS MUST ALSO SHARE THE LOAD WITH THAT PERSON IN A WAY THAT MAXIMIZES

OVERALL PERFORMANCE. IF THIS IS TO BE DONE, THE USUAL ENGINEERING SOLUTION OF HAVING THE OPERATOR TAKE CARE OF THE ODDS AND ENDS THAT ARE INCONVENIENT TO ACCOMPLISH AUTOMATICALLY WILL HAVE TO GO IN FAVOR OF GIVING HUMAN CHARACTERISTICS MAJOR CONSIDERATION.

CONSEQUENTLY, IT IS IMPORTANT TO UNDERSTAND SOMETHING OF THE HUMAN INFORMATION PROCESSING LIMITATIONS, BIASES AND PREDELICTIONS THAT AUTOMATED DECISION MAKING AND PROBLEM SOLVING SYSTEMS WILL HAVE TO CONTEND WITH. THE VIEWPOINT THAT WILL BE TAKEN IN DOING THIS WILL BE THAT OF SIMON'S (1957) CONCEPT OF BOUNDED RATIONALITY. THIS VIEW IS ESSENTIALLY THAT BECAUSE OF THEIR VERY LIMITED INFORMATION PROCESSING CAPABILITY, PEOPLE CAN ACT RATIONALLY ONLY WITH RESPECT TO MODELS OF THE WORLD THAT ARE EXTREME SIMPLIFICATIONS OF REALITY. THIS VIEW ENABLES US TO FOCUS ON THOSE LIMITS WHICH AUTOMATED SYSTEMS SHOULD AUGMENT AND WITH WHICH THEY MUST INTERACT. THERE IS ANOTHER SIDE TO THE NOTION OF BOUNDED RATIONALITY THAT IS OFTEN NEGLECTED WHEN ONE CONSIDERS HUMAN PERFORMANCE, AND THAT IS THE EXCEEDINGLY RICH AND SUBTLE WEB OF INTENTIONS, EXPECTATIONS, AND VALUES WITHIN WHICH THAT RATIONALITY, HOWEVER BOUNDED, IS EXERCISED.

2. HUMAN INFORMATION PROCESSING CHARACTERISTICS AND LIMITATIONS

2.1 SINGLE CHANNEL PROCESSOR:

FOR ALL PRACTICAL PURPOSES, ONE CAN ATTEND TO ONLY ONE THING AT A TIME. ACTIONS CAN BE CARRIED OUT IN PARALLEL IF THEY DO NOT INTERACT TOO STRONGLY. JUST AS WHEN DRIVING ONE CAN

DOWNSHIFT ON A TURN, DECISION MAKING AND ACTION SELECTION AT THE CONSCIOUS LEVEL CAN ONLY BE DONE IN SERIES. WHEN INPUTS COME CLOSE TOGETHER IN TIME, THE FIRST MUST GENERALLY BE PROCESSED COMPLETELY BEFORE THE SECOND CAN BEGIN TO BE ACTED UPON (WELFORD, 1968). PERCEPTUAL PROCESSES THAT REQUIRE ACTIVE INTERPRETATION MUST ALSO BE MULTIPLEXED IF A NUMBER OF INPUT SOURCES ARE TO BE MONITORED. ONE MAY BE ABLE TO DO THIS QUITE RAPIDLY, BUT IF THE NUMBER OF SOURCES IS LARGE, OR IF THEY ARE NOT WELL INTEGRATED, THERE MAY BE A SIGNIFICANT PENALTY IN SWITCHING RATE.

THERE IS A REMARKABLE ABILITY TO DISCRIMINATE AMONG DIFFERENT SOURCES OF INPUT EVEN WHEN THEY ARE COMPETING FOR THE SAME SENSORY CHANNEL, AS IS EVIDENT FROM THE FACT THAT ONE CAN CONVERSE AT A LOUD PARTY. EVEN IF ONE IS ATTENDING TO ONE INPUT SOURCE EXCLUSIVELY, OTHER SOURCES CAN INTERRUPT AND CATCH ONE'S ATTENTION. THIS, OF COURSE, IS BOTH AN ADVANTAGE AND A SOURCE OF ERRORS AND ACCIDENTS.

2.2 VIGILANCE:

ONE OFTEN HAS TO ATTEND "FOR" RATHER THAN "TO" SOMETHING, AND IN THIS CASE THE HIGHLY SELECTIVE ATTENTIONAL MECHANISM MAY HAVE NO EXTERNAL STIMULUS ON WHICH TO FOCUS. IN CASES OF THIS KIND PERFORMANCE CAN BE UNRELIABLE, AND IS GENERALLY FOUND TO CHANGE OVER TIME (BROADBENT & GREGORY, 1963). THIS MAY BE DUE TO BOREDOM, DISTRACTION, AND THE LIKE, WHICH REDUCE ONE'S ABILITY TO DETECT SIGNALS. BUT THERE IS ALSO A CURIOUS CHANGE IN ONE'S CRITERION FOR DECIDING THAT A SIGNAL HAS BEEN RECEIVED; ONE TENDS TO REQUIRE STRONGER EVIDENCE AS TIME ELAPSES WITH VERY FEW

SIGNALS HAVING BEEN OBSERVED. THIS "VIGILANCE DECREMENT" CAN BE OVERCOME BY ARTIFICIALLY INCREASING THE SIGNAL FREQUENCY.

2.3 LIMITED SHORT TERM MEMORY SPAN:

THE AMOUNT THAT CAN BE KEPT IN THE SHORT TERM OR WORKING MEMORY IS ONLY ABOUT 8 ITEMS (MILLER, 1956) UNLESS THERE IS STRONG SEQUENTIAL CONSTRAINT, AS THERE IS IN LANGUAGE, IN WHICH CASE THE AMOUNT OF INFORMATION THAT CAN BE HELD IN STORE IS ROUGHLY CONSTANT. (MARKS & JACK, 1952). SINCE, WITH RANDOM SEQUENCES, THE INFORMATION PER ITEM IS NOT IMPORTANT, SMITH (CITED IN MILLER, 1956) LEARNED TO TRANSLATE BINARY TO DECIMAL DIGITS VERY RAPIDLY AND BY THIS MEANS COULD ALMOST TRIPLE HIS SPAN FOR BINARY DIGITS, THE NUMBER OF DIGITS HE COULD HEAR AND REPEAT BACK IMMEDIATELY AFTERWARD.

2.4 LIMITED PROCESSING RATE:

THE RATE AT WHICH ONE CAN TRANSMIT INFORMATION IN SENSORY-MOTOR TASKS, SUCH AS COMMUNICATING WITH A COMPUTER, IS IN THE NEIGHBORHOOD OF ONLY 10 BITS/SECOND, ALTHOUGH DIFFERENT TASKS HAVE DIFFERENT MAXIMUM RATES (SHERIDAN & FERRELL, 1974). THE HIGHEST TRANSMISSION THAT HAS BEEN MEASURED FOR A KEYING TASK IS ABOUT 16 BITS/SECOND FOR PIANO PLAYING, AND THAT WAS FOR RANDOM MUSIC! JUDGMENTS THAT CATEGORIZE INPUTS ARE LIMITED IN THE AMOUNT OF INFORMATION THEY CAN CONVEY IF THERE IS NO OPPORTUNITY FOR COMPARISON WITH A STANDARD. CATEGORIZING ON A SINGLE DIMENSION CAN TRANSMIT ONLY SEVERAL BITS, EVEN WITH HIGHLY DISCRIMINABLE STIMULI SUCH AS DIFFERENT HUES. BUT WHEN COMPARISON

IS POSSIBLE, DISCRIMINATION IS EXCEEDINGLY GOOD, MUCH BETTER THAN WHEN CATEGORIES MUST BE JUDGED ABSOLUTELY.

2.5 SOURCES OF PERFORMANCE ERRORS:

PERFORMANCE ERRORS ARE HERE INTENDED TO MEAN THOSE THAT DO NOT INVOLVE BASIC MISCONCEPTIONS, BIASES, OR OVERSIGHTS. THEY RESULT LARGELY FROM TASK DEMANDS EXCEEDING THE BASIC INFORMATION PROCESSING CAPABILITIES OF THE PERSON, OR FROM TASK AND SITUATIONAL FACTORS THAT ARE SUFFICIENTLY MISMATCHED WITH HIS SENSORY-MOTOR AND PERCEPTUAL CHARACTERISTICS.

AN IMPORTANT FORM OF MISMATCH, WHICH HAS ITS ANALOG IN HIGHER LEVEL COGNITIVE FUNCTIONING AS WELL, IS THAT IN WHICH THE RESPONSES A PERSON HAS LEARNED TO ASSOCIATE WITH PARTICULAR STIMULI ARE, BECAUSE OF THE WAY THE TASK OR EQUIPMENT IS STRUCTURED, NO LONGER THE RIGHT ONES. THIS STIMULUS-RESPONSE INCOMPATIBILITY RESULTS IN CLOSING VALVES WHEN ONE WOULD OPEN THEM, MISREADING INDICATORS, USING IMPROPER COMMANDS IN COMPUTER SYSTEMS, OPERATING THE WRONG CONTROLS IN AIRCRAFT, ETC. SUCH ERRORS CAN BE REDUCED BY PROPER DESIGN.

INFORMATION OVERLOAD IS THE RESULT OF THE TASK REQUIRING A HIGHER PROCESSING RATE THAN ONE CAN ACHIEVE. WHEN SUCH CONDITIONS ARE MOMENTARY, IT MAY BE POSSIBLE TO COPE WITH THEM WITHOUT SERIOUSLY COMPROMISING PERFORMANCE. ONE MAY BE ABLE TO STORE INPUTS IN MEMORY FOR LATER PROCESSING OR COMBINE ACTIONS IN SUCH A WAY THAT TWO CAN BE CARRIED OUT AT ONCE. FOR THE MOST PART, HOWEVER, ONE MUST TRADE OFF RAPID RESPONSE WITH ACCURACY IN

SOME WAY, E.G. BY RESPONDING LESS PRECISELY OR WITH GREATER CHANCE OF SELECTING THE WRONG RESPONSE. IT MAY BE POSSIBLE SELECTIVELY TO ATTEND TO THOSE INPUTS THAT ARE THE MOST IMPORTANT, BUT IF DEMAND IS SUFFICIENTLY HIGH, PARTS OF THE TASK WILL BE OMITTED (MILLER, J., 1962). THE INTERNAL LOAD OF WORRY AND DAYDREAMING AND THE ENVIRONMENTAL LOAD OF DISTRACTION AND UNCOMFORTABLE CONDITIONS REDUCE THE CAPACITY ONE HAS TO PERFORM A TASK AND MAKE OVERLOAD MORE LIKELY (FERRELL, 1980).

A MUCH MORE SUBTLE FORM OF ERROR IS THAT IN WHICH A PORTION OF A SEQUENCE OF CUES AND ACTIONS THAT MAKE UP A SKILLED BEHAVIOUR OCCURS IN AN INAPPROPRIATE CONTEXT, AND THE ENTIRE BEHAVIOUR IS CARRIED OUT(NORMAN,1980). EVERYONE HAS HAD THE EXPERIENCE OF FINDING HIMSELF DOING ONE THING WHEN HE INTENDED TO DO ANOTHER. THIS PARTICULAR ERROR IS DANGEROUS, SINCE IT IS LIKELY TO OCCUR IN CIRCUMSTANCES WHERE PROCEDURES ARE IMPORTANT AND HIGHLY LEARNED. CHECK LISTS, SUCH AS ARE USED IN AVIATION, GUARD AGAINST IT.

2.6 IMPLICATIONS OF BASIC PROCESSING CHARACTERISTICS:

AUTOMATED SYSTEMS, BY REDUCING THE NEED FOR ROUTINE INTERVENTION AND INTERACTION WITH PEOPLE HAVE TENDED TO REDUCE THE FREQUENCY OF INSTANCES IN WHICH HUMAN CAPACITIES ARE EXCEEDED BY TASK DEMANDS. THAT IS ONE OF THE PRINCIPAL REASONS FOR AUTOMATION, IN ADDITION TO ACHIEVING HIGHER OVERALL PERFORMANCE. THE NATURE OF HUMAN LIMITATIONS SUGGESTS THOSE ASPECTS OF FUNCTIONING THAT IT IS MOST CRITICAL TO AUTOMATE. DATA INPUT, MEASUREMENT, SHORT TERM STORAGE OF INFORMATION, RAPID ANALYSIS OF

MULTIPLE INPUTS, AND OPERATIONS OF ALL KINDS IN WHICH SPEED OF RESPONSE IS IMPORTANT ARE ALL PRIME TARGETS FOR AUTOMATION.

THERE IS, HOWEVER, AN ASPECT OF THE PROBLEM THAT CAN BE OVERLOOKED FROM THIS STEREOTYPICAL VIEWPOINT. AS MENTIONED AT THE BEGINNING, THERE IS ALWAYS A MAN-MACHINE INTERFACE. WITH AN AUTOMATED SYSTEM, PARTICULARLY ONE THAT OPERATES IN A COGNITIVE RATHER THAN PHYSICAL MODE, ONE SHOULD BE CAREFUL NOT TO TRADE HIGH RATES OF ROUTINE ERROR BUT GLOBAL RELIABILITY FOR ROUTINE RELIABILITY BUT THE POTENTIAL FOR DISASTER. ONE DOES NOT WANT TO ALLOW THE SENSITIVITY OF THE SYSTEM TO ERROR TO INCREASE BY SHIFTING THE LEVEL AT WHICH PEOPLE INTERACT WITH IT IN A ROUTINE WAY UPWARD FROM OPERATION TO PLANNING AND STRATEGY FORMATION. A WAY TO AVOID SUCH AN INCREASE IS, OF COURSE, TO INSTITUTE PROCEDURAL CHECKS AND INSTITUTIONAL SAFEGUARDS. A MORE SATISFACTORY METHOD IS TO DESIGN THE SYSTEM SO THAT HIGHER LEVELS OF INTERACTION WITH IT ARE CARRIED OUT IN HIGHER LEVELS OF DISCOURSE. THIS WAS THE PROMISE OF HIGH LEVEL PROGRAMMING LANGUAGES, A PROMISE THAT, FOR THE ORDINARY USER, AT LEAST, REMAINS UNFULFILLED.

3. CHARACTERISTICS RELATED TO DECISION

3.1 INFERENCE AND UNCERTAINTY:

THE CHARACTERISTICS OF PEOPLE THAT RELATE TO THEIR PERCEPTION OF RANDOMNESS AND TO THE INFERENCES THEY MAKE UNDER CONDITIONS OF UNCERTAINTY ARE POTENTIALLY IMPORTANT IN AUTOMATED DECISION AND PROBLEM SOLVING SYSTEMS FOR SEVERAL REASONS. A)

PEOPLE'S SHORTCOMINGS MAY POINT THE WAY TO THOSE ASPECTS OF DECISION MAKING AND PROBLEM SOLVING THAT WOULD MOST BENEFIT FROM AUTOMATION. B) SUCH SYSTEMS WILL ALMOST CERTAINLY REQUIRE HUMAN JUDGMENTS AS INPUT. C) USERS WILL HAVE TO INTERPRET THE OUTPUT, THE ACCEPTABILITY AND USEFULNESS OF WHICH WILL DEPEND IN SOME MEASURE ON HOW IT IS PERCEIVED. D) THE PROBLEMS OF EVALUATION OF PERFORMANCE THAT CONFRONT THE RESEARCH ON HUMAN BEHAVIOR MUST BE DEALT WITH IN THE DESIGN OF AUTOMATED SYSTEMS.

ALTHOUGH PEOPLE CAN OFTEN MAKE ACCURATE JUDGMENTS ABOUT SOME ASPECTS OF DIRECTLY OBSERVED RANDOM PROCESSES (FIKE & FERRELL, 1978, PETERSON & BEACH, 1967), THERE IS AMPLE EVIDENCE INDICATING THAT VERY SERIOUS ERRORS ARE LIKELY IF PEOPLE ATTEMPT TO FUNCTION AS "INTUITIVE STATISTICIANS".

3.1.1 THE HYPOTHESIS OF RANDOMNESS:

RANDOMNESS IS AN HYPOTHESIS THAT A PERSON MAY OR MAY NOT ACCEPT, AND WHICH, IF ACCEPTED, IS LIKELY TO LEAD TO CERTAIN EXPECTATIONS ABOUT THE EVENTS IN QUESTION. ONE OF THESE EXPECTATIONS IS JOCULARLY CALLED THE LAW OF SMALL NUMBERS (TVERSKY & KAHNEMAN 1971). THE LAW OF SMALL NUMBERS IS ESSENTIALLY THE BELIEF THAT RANDOMNESS IS A PROPERTY OF EVENTS THEMSELVES AND THAT, SINCE SAMPLE SIZE IS NOT A PROPERTY OF EVENTS, IT IS IRRELEVANT IN CONSIDERATIONS OF RANDOMNESS. FOR EXAMPLE, STATISTICALLY NAIVE PEOPLE EXPECT THAT IN A RANDOM SEQUENCE IN WHICH A PARTICULAR EVENT OCCURS WITH PROBABILITY P, ALL SUBSEQUENCES SHOULD BE IRREGULAR AND SHOULD EXHIBIT THAT EVENT IN APPROXIMATELY THE PROPORTION P REGARDLESS OF THEIR LENGTH. THE BIAS IS SO STRONG THAT EVEN PSYCHOLOGISTS WITH

GRADUATE TRAINING IN STATISTICS GROSSLY UNDERESTIMATE THE SAMPLE SIZES NECESSARY FOR SIGNIFICANT DIFFERENCES WHEN THEY MUST DO SO WITHOUT BENEFIT OF CALCULATION. A DEVIATION OF A GIVEN AMOUNT FROM THE EXPECTED PROPORTION IS GENERALLY SEEN AS BEING JUST AS LIKELY REGARDLESS OF THE SAMPLE SIZE. SYSTEMS ENGINEERS WITH GRADUATE TRAINING IN PROBABILITY AND STATISTICS CAN ACCURATELY JUDGE THE STANDARD DEVIATION OF NORMAL DISTRIBUTIONS FROM SMALL SAMPLES, BUT THEIR JUDGMENTS OF THE STANDARD DEVIATIONS OF THE MEANS OF THOSE SAMPLES ARE SERIOUSLY IN ERROR BECAUSE THEY GIVE SO LITTLE WEIGHT TO SAMPLE SIZE (FIKE, 1967).

THE GAMBLER'S FALLACY IS ANOTHER PREVALENT MISCONCEPTION ABOUT THE NATURE OF RANDOM SEQUENCES. THE RUN OF AN EVENT THAT IS NOT IMMEDIATELY FOLLOWED BY A COUNTERBALANCING DECREASE IN THE FREQUENCY OF ITS OCCURRENCE APPEARS TO CONTRADICT THE CONSTANCY OF THE PROBABILITY OF THE EVENT. THE IDEA IS SO COMPELLING THAT A PROFESSOR AT A MEETING OF THE AMERICAN INSTITUTE FOR DECISION SCIENCES GAVE A PAPER EXPLAINING WHY THE GAMBLER'S FALLACY IS NOT A FALLACY.

IN GENERAL, THERE SEEMS TO BE A DISPOSITION TO PREFER THE INTERNAL UNCERTAINTY ABOUT THE CORRECTNESS OF A DETERMINISTIC SUBJECTIVE MODEL OF WHAT IS GOING ON TO THE EXTERNAL UNCERTAINTY IMPLIED IN THE MORE PARSIMONIOUS, BUT LESS INTUITIVELY APPEALING STOCHASTIC MODEL. HOWEVER, PEOPLE SEEM LITTLE BETTER ABLE TO DEAL WITH THEIR INTERNAL UNCERTAINTY THAN WITH EXTERNAL RANDOMNESS, AS SHALL BE INDICATED FURTHER ON.

PEOPLE'S ESTIMATES OF PROBABILITY ARE MOST RELIABLE IN JUST THOSE SITUATIONS IN WHICH MACHINE ESTIMATES ARE ALSO POSSIBLE,

NAMELY DIRECTLY OBSERVED SEQUENCES OR SAMPLES OF EVENTS. BUT FOR LESS WELL DEFINED SITUATIONS, SUBJECTIVE ESTIMATES WILL PROBABLY HAVE TO BE USED IF FORMAL DECISION ANALYSIS IS TO BE UNDERTAKEN. NUMEROUS, SYSTEMATIC BIASES IN SUBJECTIVE PROBABILITY HAVE BEEN FOUND. MANY OF THEM APPEAR TO RESULT FROM PLAUSIBLE BUT MISLEADING HEURISTICS BEING USED TO FACILITATE JUDGMENT. SOME OF THESE WILL BE CONSIDERED BELOW.

3.1.2 REPRESENTATIVENESS:

REPRESENTATIVENESS REFERS TO THE TYPE OF BIAS THAT OCCURS WHEN DATA IS ASSUMED TO COME FROM THE CATEGORY OF WHICH IT IS MOST REPRESENTATIVE, WITHOUT REGARD FOR THE PRIOR LIKELIHOOD OF THAT CATEGORY (KAHNEMAN & TVERSKY, 1972). IT IS A VERY COMPELLING BIAS IN THE SENSE THAT EVEN WHEN CALLED TO PEOPLE'S ATTENTION, THEY FIND IT HARD TO RECONCILE THEIR SUBJECTIVE FEELINGS WITH WHAT THEY KNOW TO BE STATISTICALLY CORRECT. FOR EXAMPLE, IF THE PRIOR PROBABILITY OF A CONDITION IS .1 AND A TEST OF WHETHER OR NOT THIS CONDITION HOLDS IS CORRECT 90% OF THE TIME, IT SEEMS NOT TO BE INTUITIVELY SATISFYING TO HAVE TO CONCLUDE THAT THE PROBABILITY OF THE CONDITION GIVEN A POSITIVE TEST RESULT IS ONLY .5. THE PRIOR SEEMS IRRELEVANT IN VIEW OF THE TEST.

THERE IS AN INTERESTING QUALIFICATION TO THIS RESULT. IF THE DATA THAT DETERMINES THE PRIOR CAN BE INTERPRETED CAUSALLY RATHER THAN SIMPLY DIAGNOSTICALLY, IT IS GIVEN MORE WEIGHT (TVERSKY & KAHNEMAN, 1977).

3.1.3 AVAILABILITY:

WHEN JUDGING THE LIKELIHOOD OF EVENTS THAT CANNOT BE

DIRECTLY OBSERVED, ONE MAY SAMPLE ONE'S MEMORY FOR THEM AND JUDGE THE LIKELIHOOD IN RELATION TO THE EASE OF RECALL OR BY THE RELATIVE NUMBER OF INSTANCES THAT CAN BE BROUGHT TO MIND. FOR EXAMPLE, IF ONE WERE TO ASSESS THE RELATIVE LIKELIHOOD OF A RANDOMLY CHOSEN WORD BEGINNING WITH K AS OPPOSED TO HAVING K AS ITS THIRD LETTER, ONE MIGHT ATTEMPT TO GENERATE SAMPLES OF SUCH WORDS AND THIS COULD EASILY LEAD TO THE ERRONEOUS CONCLUSION THAT WORDS BEGINNING WITH K ARE MORE LIKELY (TVERSKY & KAHNEMAN, 1973A). THE MANNER IN WHICH INFORMATION IS STORED IN MEMORY, THE WAY IN WHICH IT IS ACCESSED, AND SUCH ASPECTS AS SALIENCY, EMOTIONAL CONNOTATION, AND ASSOCIATION CAN BIAS THE RECALL OF EVENTS OR THE PRODUCTION OF INSTANCES. THIS MAY ALSO HELP EXPLAIN WHY DESIRABLE EVENTS ARE OFTEN GIVEN HIGHER PROBABILITY THAN UNDESIRABLE, AND EVENTS RECENTLY CONSIDERED SEEM MORE LIKELY TO OCCUR.

3.1.4 CALIBRATION AND OVERCONFIDENCE:

CALIBRATION IS THE EXTENT TO WHICH THE FREQUENCIES OF OCCURRENCE OF EVENTS MATCH THE SUBJECTIVE PROBABILITIES ASSIGNED TO THEM. IT CAN BE ASSESSED FOR SETS OF EVENTS THAT ARE NOT REPEATABLE, BUT ONE MUST BE CAREFUL THAT THE EVENTS CONSTITUTE AN EXCHANGEABLE SET. THE GENERAL FINDING IS THAT OF OVERCONFIDENCE, ASSIGNMENT OF PROBABILITIES THAT ARE TOO EXTREME (LICHTENSTEIN & FISCHHOFF, 1977). IN ADDITION, THE DEGREE OF OVERCONFIDENCE DEPENDS ON THE OVERALL PROPORTION OF OCCURRENCE. IN THE TYPICAL EXPERIMENT ONE GIVES HIS SUBJECTIVE PROBABILITY OF A CORRECT CHOICE, AND IN THIS CASE THE DEGREE OF OVERCONFIDENCE IS LESS THE HIGHER THE PROPORTION OF CORRECT CHOICES. INDEED, WHEN IT IS

HIGH ENOUGH THE CALIBRATION SHOWS UNDERCONFIDENCE. A MODEL ACCOUNTING FOR THIS EFFECT IN TERMS OF A FAILURE TO CHANGE THE RESPONSE CRITERIA IN ACCORDANCE WITH THE DIFFICULTY OF THE ITEMS HAS BEEN DEVELOPED (FERRELL & MCGOEY,1980). CALIBRATION ACCURACY CAN BE MUCH IMPROVED WITH FEEDBACK (LICHTENSTEIN & FISCHHOFF, 1979). WEATHER FORECASTERS ARE REASONABLY WELL CALIBRATED.

THERE CAN BE NO FUNCTION CHARACTERISTIC OF AN INDIVIDUAL THAT RELATES SUBJECTIVE PROBABILITY TO OBJECTIVE RELATIVE FREQUENCY, SINCE THE EXISTENCE OF SUCH A FUNCTION, OTHER THAN THE IDENTITY FUNCTION, WOULD REQUIRE THAT SUBJECTIVE PROBABILITIES NOT SUM TO ONE FOR MUTUALLY EXCLUSIVE AND EXHAUSTIVE EVENTS. THUS CALIBRATION IS PECULIAR TO THE TASK AND IS IN RELATION TO THE SET OF JUDGMENTS THAT ARE MADE.

CALIBRATION CAN BE ASSESSED FOR CONTINUOUS VARIABLES, TOO, AND OVERCONFIDENCE IS COMMON IN THAT CASE. ONE INFERS THE SUBJECTIVE PROBABILITY DENSITY FUNCTION BY FINDING THE PROPORTION OF INSTANCES IN WHICH THE CORRECT VALUE FELL WITHIN SUBJECTIVELY GIVEN INTERVALS OF THE VARIABLE CORRESPONDING TO STATED INTERFRACTILE RANGES. USUALLY THE VARIANCE OF THE SUBJECTIVE PDF IS TOO SMALL, INDICATIVE OF OVERCONFIDENCE. IT HAS BEEN PROPOSED THAT THIS RESULTS FROM A HEURISTIC WHEREBY ONE ANCHORS ON HIS BEST GUESS FOR THE VALUE OF THE VARIABLE AND THEN ADJUSTS UP AND DOWN TO OBTAIN THE FRACTILES (TVERSKY & KAHNEMAN, 1973B). THE ADJUSTMENT TENDS TO BE INSUFFICIENT, PERHAPS BECAUSE THE VARIETY OF POSSIBLE SOURCES OF UNCERTAINTY IS NOT RECOGNIZED.

3.1.5 OTHER CHARACTERISTICS OF PROBABILISTIC INFERENCE:

IN SEQUENTIAL PROBABILITY REVISION IN WHICH THE

PROBABILITY OF AN HYPOTHESIS IS UPDATED ON THE BASIS OF EVIDENCE, IT IS COMMONLY FOUND THAT THE REVISION IS LESS EXTREME THAN IS ACTUALLY WARRANTED. THIS CONSERVATISM, AS IT IS KNOWN, IS IN PART AN ARTIFACT OF THE LIMITED SCALE OF PROBABILITY, BUT IT ALSO IS DUE TO INABILITY OF THE PERSON ACCURATELY TO COMBINE THE IMPACTS OF THE EVIDENCE EVEN THOUGH HE IS AN EXCELLENT JUDGE OF WHAT EVIDENCE IS RELEVANT (EDWARDS, 1968). A SIMILAR INABILITY TO COMBINE DATA OPTIMALLY IS FOUND IN COMPARISONS OF PEOPLE'S JUDGMENT WITH LINEAR REGRESSION MODELS (DAWES & CORRIGAN, 1974). OPTIMAL MODELS ARE FAR SUPERIOR TO HUMAN JUDGMENT WITH THE SAME EVIDENCE, AND MODELS BASED ON THE JUDGE'S OWN JUDGMENTS ARE MORE RELIABLE THAN THE JUDGE.

3.1.6 IMPLICATIONS OF PEOPLE'S PROBABILISTIC INFORMATION PROCESSING CHARACTERISTICS FOR AUTOMATED DECISION AND PROBLEM SOLVING SYSTEMS:

ON THE WHOLE, THE CURRENT FASHION IN EXPERIMENTS THAT SUGGEST THAT PEOPLE ARE ILL-EQUIPPED TO DEAL INTUITIVELY WITH UNCERTAINTY IN A STATISTICALLY SOUND MANNER STRONGLY SUGGESTS AUTOMATING AS MUCH OF THEIR PERFORMANCE IN THIS AREA AS POSSIBLE. THE CURRENT WORK TOWARD AUTOMATING THE PRODUCTION OF PROBABILISTIC WEATHER FORECASTS IS AN INDICATION OF WHAT MAY BE DONE ALONG THIS LINE IN A WELL DEFINED (THOUGH EXCEEDINGLY COMPLEX) PROBLEM.

ALTHOUGH PEOPLE'S PERFORMANCE APPEARS TO BE VERY POOR WHEN COMPARED WITH THE OPTIMAL BEHAVIOR IN CLEAR-CUT EXPERIMENTS, IT SHOULD BE REMEMBERED THAT PEOPLE DEAL WITH UNCERTAINTY IN FACT, EVEN IF VERY SUBOPTIMALLY BY SOME STANDARDS. THE TECHNIQUES BY

WHICH THEY DEAL WITH UNCERTAINTY APPEAR TO BE ONES THAT PRODUCE ADEQUATELY RATIONAL BEHAVIOR IN THE FACE OF VERY LITTLE INFORMATION AND A RELATIVELY SMALL AMOUNT OF COMPUTING CAPACITY AND IN THE PRESENCE OF VERY LARGE UNCERTAINTIES ABOUT THE STRUCTURE OF THE PROBLEMS BEING DEALT WITH. AUTOMATION ALLOWS REMEDYING THE COMPUTATIONAL LIMITATION TO SOME EXTENT, AND POSSIBLY ALLEVIATING THE LACK OF INFORMATION SOMEWHAT, BUT THE UNCERTAINTIES ABOUT THE STRUCTURE OF THE PROBLEMS BEING DEALT WITH REMAIN. IT WOULD NOT BE SURPRISING TO FIND THAT AUTOMATED DECISION SYSTEMS EVOLVE INTO ONES THAT EXHIBIT THE SAME KINDS OF SUBOPTIMALITY THAT ONE FINDS IN EXPERIMENTS WITH PEOPLE.

IF THIS IDEA HAS ANY VALIDITY, IT MAY BE WORTH WHILE TO ATTEMPT TO STUDY THE WAYS IN WHICH PEOPLE CAN FORMULATE ROBUST STRATEGIES IN THE PRESENCE OF UNCERTAINTY WITH THE OBJECTIVE OF IMPLEMENTING SUCH TECHNIQUES IN AUTOMATED SYSTEMS.

PEOPLE MUST INTERPRET THE OUTPUT OF AUTOMATED OR SEMI-AUTOMATED SYSTEMS. TO THE USER, THE SYSTEM IS A SOURCE OF UNCERTAIN INFORMATION AND POSES THE NEED TO MAKE DECISIONS WITH RESPECT TO IT. HIS RESPONSE TO IT WILL EXHIBIT THE SAME KINDS OF BIAS AND SUBOPTIMALITY SHOWN IN HIS OTHER RESPONSES TO UNCERTAINTY. THIS SHOULD HIGHLIGHT THE IMPORTANCE OF ESTABLISHING A SUITABLE LEVEL OF COMMUNICATION WITH SUCH DEVICES.

3.2 UTILITY AND VALUE:

THERE ARE TWO KINDS OF EVALUATION THAT NEED TO BE CONSIDERED, 1) MEANS END ANALYSIS IN WHICH EVALUATION IS CONCERNED WITH THE EFFECTIVENESS OF DIFFERENT PROCEDURES OR

SOURCES OF ACTION IN ACHIEVING A GOAL, AND 2) THE EVALUATION OF OUTCOMES WITH RESPECT TO THEIR DESIRABILITY IN THE USUAL DECISION THEORETIC SENSE.

3.2.1 EVALUATION OF MEANS END RELATIONSHIPS:

IN SKILLED MANUAL AND COGNITIVE PERFORMANCE PEOPLE ARE EXTREMEMLY CAPABLE OF EVALUATING THE RELATIVE EFFECTIVENESS OF PROCEDURES AND ACTIONS, AND THEY DO SO WITH GREAT SPEED. MANIPULATION, THE USE OF TOOLS, CHESS PLAYING, CONTROL OF AIRCRAFT ARE GOOD EXAMPLES. MACHINES CAN DO ALL OF THESE THINGS, THE LATTER BETTER THAN PILOTS CAN, BUT THERE IS NO QUESTION OF THE PREEMINENCE OF THE HUMAN IN THE MORE COMPLEX ENVIRONMENTS. THE BASIS SEEMS TO BE THE HIGHLY INTEGRATED STIMULUS- RESPONSE ASSOCIATIONS IN MEMORY THAT TAKE MANY YEARS AND MUCH PRACTICE TO DEVELOP, AND THAT OPERATE BELOW, OR NOT MUCH ABOVE THE LEVEL OF CONSCIOUSNESS FOR THE MOST PART.

THE IMPLICATION FOR AUTOMATED SYSTEMS IS NOT CLEAR, EXCEPT THAT PERHAPS A SIMILAR APPROACH MAY BE NECESSARY IF VERY RAPID SKILLED PERFORMANCE IS DESIRABLE, AS IT IS WITH MANIPULATION. ALBUS & EVANS (1975) HAVE WORKED TOWARD THIS END USING A MODEL OF THE CEREBELLUM TO ATTEMPT TO AVOID THE EXCEEDINGLY HIGH COMPUTATIONAL COSTS OF GEOMETRICAL ANALYSIS NEEDED TO DECIDE HOW TO CARRY OUT MANIPULATIONS.

THE STUDIES OF MEANS END ANALYSIS IN TASKS IN WHICH SENSORY MOTOR SKILLS FIGURE LESS AND AD HOC REASONING IS OF MORE IMPORTANCE, SUCH AS TROUBLE SHOOTING, INDICATE THAT PERFORMANCE IS VERY MUCH DEPENDENT UPON AVAILABLE INFORMATION. WHEN ONE HAS TO DERIVE POSSIBLE PROCEDURES WITHOUT ASSISTANCE, LIKELY

PROSPECTS ARE OVERLOOKED, PROPERTIES OF THE DEFECTIVE SYSTEM ARE IGNORED AND THE LIMITS OF ONE'S INFORMATION PROCESSING CAPACITIES BECOME EVIDENT. TROUBLE SHOOTERS ADOPT HEURISTICS TO ASSIST THEIR JUDGMENT, AND THEY ARE GENERALLY SUBOPTIMAL, BUT OFTEN ADEQUATE (ROUSE, 1978).

THE IMPLICATIONS HERE ARE THAT AUTOMATION OF THIS KIND OF EVALUATION AND DECISION TASK IS LIKELY TO BE WORTHWHILE, EXPECIALLY WHEN THE MODELS OF THE PROCESS ARE CLEAR CUT. AUTOMATED SYSTEMS FOR AIDING THE OPERATOR AND FOR TRAINING HIM ARE ACTIVELY BEING DEVELOPED.

3.2.2 UTILITY:

UTILITY THEORY ALLOWS ONE TO DETERMINE THE RELATIVE PREFERENCE THAT SHOULD BE ACCORDED CONSEQUENCES THAT WOULD OTHERWISE BE INCOMMENSURABLE. IT IS A PRESCRIPTIVE THEORY, NOT A DESCRIPTIVE ONE, AT LEAST NOT A VERY POWERFUL DESCRIPTIVE ONE. THERE IS AMPLE EVIDENCE THAT ACTUAL PREFERENCE STRUCTURES FREQUENTLY FAIL TO SATISFY THE AXIOMS OF UTILITY THEORY, AND THAT KNOWING WHAT ONE PREFERS IS MORE OF A K DYNAMIC PROCESS OF SEARCH AND RECONSIDERATION THAN A SIMPLE STATE OF KNOWLEDGE.

IN COMPLEX DECISION SITUATIONS, THE ROLE OF UTILITY THEORY IS TO PROVIDE A MEANS FOR ARRIVING AT A DECISION. ONE MAY BE WILLING TO ACCEPT IT FOR THIS PURPOSE, EVEN IF IT DOES NOT PERMIT AN ACCURATE REPRESENTATION OF ONE'S PREFERENCE STRUCTURE. THIS IN ITSELF IS A DECISION OF SOME MOMENT AND SHOULD BE RECONSIDERED ANEW IN EACH CASE.

A CATALOG OF THE WAYS IN WHICH UTILITY THEORY CAN FAIL TO

ACCOUNT FOR PREFERENCE WOULD BE VERY LONG, BUT A FEW OF THE PROBLEMS CAN BE MENTIONED.

1. PROBABILITIES AND VALUES ARE OFTEN NOT INDEPENDENT IN A PERSON'S IMPLICIT VIEW (SLOVIC, 1966).

2. PREFERENCES FOR THE VARIANCE, OR EVEN THE SKEW OF PROBABILISTIC ALTERNATIVES ARE OFTEN FOUND (VANDERMEER, 1963).

3. PEOPLE FREQUENTLY ARE UNABLE TO MAKE CONSISTENT COMPARISONS BETWEEN SURE THINGS AND BETS.

4. UTILITIES FOR A GIVEN COMMODITY ARE OFTEN FOUND TO BE DIFFERENT DEPENDING ON HOW THE UTILITY IS ASSESSED.

5. MULTIDIMENSIONAL ALTERNATIVES ARE OFTEN INTRANSITIVELY VALUED, ESPECIALLY IF THE COMBINATIONS OF ADVANTAGES OF THE DIFFERENT DIMENSIONS CAN BE VIEWED FROM DIFFERENT PERSPECTIVES (TVERSKY, 1969).

6. ASSESSMENT OF THE UTILITY OF ONE OUTCOME MAY DEPEND ON THE RANGE OF OTHER OUTCOMES WITH WHICH IT IS COMPARED (KRZYSZTOFOWICZ & DUCKSTEIN, 1980).

3.2.3 IMPLICATIONS OF BEHAVIOR WITH RESPECT TO UTILITY:

INSOFAR AS AUTOMATED SYSTEMS DEAL WITH DECISIONS AND PROBLEMS OF A RELATIVELY VALUE FREE SORT, OR DEAL WITH ONES ON WHICH THERE IS UNIVERSAL AGREEMENT ABOUT THE ORDERINGS AND IMPORTANCE OF THE CONSEQUENCES, THE ANOMALIES IN UTILITY THEORY WILL NOT MATTER. BUT WHEN THEY DEAL WITH QUESTIONS OVER WHICH THERE IS DISPUTE, OR WHEN OUTCOMES DEPEND ON THE INPUT VALUES IN AN IMPORTANT AND COMPLEX WAY SO THAT THE RELATION IS NOT OBVIOUS, THEN IT IS LIKELY THAT ANALYSIS BEYOND THE LEVEL OF UTILITY WILL BE NECESSARY ADEQUATELY TO INFORM THE AUTOMATED SYSTEM.

ULTIMATELY DECISIONS HAVE TO BE MADE, AND PROBLEMS RESOLVED, IF NOT SOLVED. BUT THE PROSPECT OF CREATING THE GOLEM REFERRED TO BY WIENER, THE DECISION MAKING MACHINE THAT MAKES DECISIONS WITH WHICH WE DISAGREE--BUT TOO LATE, IS NOT A HAPPY ONE.

4. CHARACTERISTICS RELATED TO PROBLEM SOLVING

4.1 HUMAN-COMPUTER SYMBIOSIS, WHAT HAPPENED TO IT?

THE SCOPE OF THE QUESTION OF HUMAN CAPACITIES FOR PROBLEM SOLVING ENCOMPASSES MOST OF PSYCHOLOGY, AND CANNOT BE CONSIDERED IN DETAIL. IT IS INSTRUCTIVE TO CONSIDER THE "MAN-COMPUTER SYMBIOSIS" ENVISIONED BY J.C.R. LICKLIDER MORE THAN 20 YEARS AGO (LICKLIDER, 1960). IN THAT SEMINAL PAPER HE ENVISIONED THAT ONE SHOULD BE ABLE "TO THINK IN INTERACTION WITH A COMPUTER IN THE SAME WAY THAT YOU THINK WITH A COLLEAGUE WHOSE COMPETENCE SUPPLEMENTS YOUR OWN..." THERE ARE A GREAT MANY PEOPLE WHO USE COMPUTERS MUCH OF THEIR TIME, BUT ARE THERE ANY WHO WOULD CHARACTERIZE THE INTERACTION IN THOSE TERMS? WHY?

THERE ARE AT LEAST TWO IMPORTANT PARTS TO THAT ANSWER, AND THEY BOTH SUGGEST THAT IF THERE WAS ANY LACK OF UNDERSTANDING IN 1960 IT WAS MORE ABOUT THE NATURE OF DIALOGS WITH ONE'S COLLEAGUES THAN ABOUT WHAT COMPUTERS COULD POTENTIALLY DO.

1. THE PROPOSAL WAS, ESSENTIALLY, THAT THE COMPUTER WOULD DO WHAT IT DOES BEST, NAMELY FAST COMPUTATION AND SEARCHING AND STORAGE OF INFORMATION, THE ROUTINE ASPECTS OF INTELLECTUAL WORK. THIS IS A GOOD IDEA, BUT IT TURNS OUT TO BE FEASIBLE FOR HIGH

LEVEL PROBLEMS ONLY IF EITHER THE USER IS VERY HIGHLY SKILLED AT USING A QUITE COMPLEX SYSTEM OR IF THE MACHINE CAN COMMUNICATE AT THE LEVEL OF THE PROBLEM RATHER THAN AT THE LEVEL OF WHAT IT DOES BEST. SINCE IT IS NOT KNOWN HOW TO ACCOMPLISH THE LATTER, AT LEAST VERY WELL, THE FORMER IS THE CURRENT VERSION OF SYMBIOSIS.

2. LICKLIDER'S CONCEPTION OF SYMBIOSIS REQUIRED THE MACHINE TO BE EXCEEDINGLY GENERAL IN ITS CAPABILITIES, YET STILL AVOID HAVING THE USER NEED TO CONCERN HIMSELF WITH THE MECHANICS OF HOW THE PROGRAMS WORKED AND COULD BE USED. VERSATILITY IN COLLEAGUES IS RARE ENOUGH, AND IN INTELLIGENT ARTIFACTS IT IS LIKELY TO BE VERY SUPERFICIAL. THE IMPORTANT ADVANCES IN EXTENDING THE SCOPE OF COMPUTER APPLICATIONS HAVE IN LARGE MEASURE DEPENDED UPON EXTREMELY CAREFUL ANALYSIS OF THE TASKS TO BE DONE. QUITE STRUCTURED TASKS THAT WILL BE CARRIED OUT REPEATEDLY CAN, THUS, BE DONE QUITE INTELLIGENTLY BY MACHINE, BUT ILL-STRUCTURED, AD HOC TASKS MUST RELY ON THE OPERATOR'S DIRECTION AND SUPERVISION.

THE TWO REQUIREMENTS, THEN, ARE THAT THE SYSTEM BE ABLE TO COMMUNICATE AND TO PLAN AND ORGANIZE ITS BEHAVIOUR AT THE HIGHEST, OR PROBLEM LEVEL. THIS SEEMS TO IMPLY THAT IT MUST BE ABLE TO THINK ABOUT ITSELF IF THE IDEAL OF MAN-COMPUTER SYMBIOSIS IS TO BE REALIZED IN THE FORM ENVISIONED BY LICKLIDER.

4.2 RECOMMENDATIONS:

SEVERAL GENERAL RECOMMENDATIONS ARE SUGGESTED BY THE FOREGOING. AUTOMATED AND SEMI-AUTOMATED OR INTERACTIVE SYSTEMS WILL PROBABLY BE MOST EFFECTIVE IF THEY ARE SPECIAL PURPOSE, AND

IF THE PURPOSE AND THE TASK ARE VERY CAREFULLY ANALYZED WITH A VIEW TOWARD MAXIMUM MAN-MACHINE COMMUNICATION AT A LEVEL AT WHICH THE TASK HAS MEANING FOR THE USER. PEOPLE STRUCTURE THEIR EXPERIENCE ACCORDING TO MEANING, AND ORGANIZE THEIR MEMORIES AND THEIR PLANS AROUND INTERPRETATIONS OF THE TASK THAT INVOLVE HIGH LEVELS OF GENERALITY AS WELL AS SPECIFIC PROCEDURES. OPTIMALITY IN PROBLEM SOLVING, AS IN DECISION MAKING, IS LIKELY TO PROVE LESS IMPORTANT THAN ROBUSTNESS. FINALLY, THE HUMAN IS FALLIBLE, PRONE TO MISPERCEPTION, OVERLOAD, ERROR, AND GENERAL BUMBLING. THE SYSTEM SHOULD NOT TRUST HIS PERFORMANCE--BUT IT SHOULD PAY STRICT ATTENTION TO HIS HUNCHES.

5. REFERENCES

1. ALBUS, J.S., & EVANS, J.M., A HEIRARCHICAL STRUCTURE FOR ROBOT CONTROL. PROCEEDINGS OF THE 5TH INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ROBOTS, CHICAGO, ILL., 231-238, SEPT., 1975.

2. BROADBENT, S.E. & GREGORY, M., VIGILANCE CONSIDERED AS A STATISTICAL DECISION. BRIT. J. PSYCHOL., 65, 298-300, 1963.

3. DAWES, R.M. & CORRIGAN, B., LINEAR MODELS IN DECISION MAKING, PSYCHOL. BULL.,81, 95-106, 1974.

4. EDWARDS, W., CONSERVATISM IN HUMAN INFORMATION PROCESSING. IN KLEINMUTZ, B. (ED.), FORMAL REPRESENTATION OF HUMAN JUDGMENT. NEW YORK: WILEY, 1968.

5. FERRELL, W.R., LOAD AND CAPACITY: BEHAVIORAL ASPECTS OF SAFETY. PROCEEDINGS OF THE 38TH ANNUAL CONGRESS ON OCCUPATIONAL HEALTH, AMERICAN MEDICAL ASSOCIATION, SEPTEMBER 1978.

6. FERRELL, W.R. & MCGOEY, P., A MODEL OF CALIBRATION FOR SUBJECTIVE PROBABILITIES. ORGANIZATIONAL BEHAVIOR AND HUMAN PERFORMANCE, 26, 32-53, 1980.

7. FIKE, R.L., SUBJECTIVE ESTIMATION OF VARIABILITY. UNPUBLISHED DOCTORAL DISSERTATION, THE UNIVERSITY OF ARIZONA, 1977.

8. FIKE, R.L. & FERRELL, W.R., SUBJECTIVE DETECTION OF DIFFERENCES IN VARIANCE FROM SMALL SAMPLES. ORGANIZATIONAL BEHAVIOR AND HUMAN PERFORMANCE, 22, 262-278, 1978.

9. KAHNEMAN, D. & TVERSKY, A., SUBJECTIVE PROBABILITY: A JUDGMENT OF REPRESENTATIVENESS. COGNITIVE PSYCHOL., 430-454,1972.

10. KRZYSZTOFOWICZ, R. & DUCKSTEIN, L., ERRORS IN MULTIATTRIBUTE UTILITY ASSESSMENT. ORGANIZATIONAL BEHAVIOR AND HUMAN PERFORMANCE, 1980 (IN PRESS).

11. LICHTENSTEIN, S. & FISCHHOFF, B., DO THOSE WHO KNOW KNOW MORE ABOUT HOW MUCH THEY KNOW? ORGANIZATIONAL BEHAVIOR AND HUMAN PERFORMANCE, 20, 159-183,1977.

12. LICHTENSTEIN, S. & FISCHHOFF, B., TRAINING FOR CALIBRATION. UNPUBLISHED MS., 1979.

13. LICKLIDER, J.C.R., MAN-COMPUTER SYMBIOSIS. IRE TRANSACTIONS ON HUMAN FACTORS AND ELECTRONICS, HFE-1, 4-11, MARCH 1960.

14. MARKS, M. & JACK, O., VERBAL CONTEXT AND MEMORY SPAN FOR MEANINGFUL MATERIAL. AMER. J. PSYCHOL., 65, 298-300, 1952.

15. MILLER, G.A., THE MAGIC NUMBER SEVEN PLUS OR MINUS TWO. PSYCHOL. REV., 63, 81-97, 1956.

16. MILLER, J., INFORMATION INPUT OVERLOAD. PROCEEDINGS OF THE CONFERENCE ON SELF-ORGANIZING SYSTEMS. 1962.

17. NORMAN, D., HUMAN ERROR. PSYCHOL. TODAY, APRIL, 1980.

18. PETERSON, C.R.& BEACH, L.R., MAN AS AN INTUITIVE STATISTICIAN. PSYCHOL. BULL.,68, 29-46,1967.

19. ROUSE, W.B., A MODEL OF HUMAN DECISION MAKING IN A FAULT DIAGNOSIS TASK. IEEE TRANS. ON SYSTEMS, MAN, AND CYBERNETICS, SMC-8, 357-361, 1978.

20. SHERIDAN, T.B. & FERRELL, W.R., MAN-MACHINE SYSTEMS: INFORMATION, CONTROL, AND DECISION MODELS OF HUMAN PERFORMANCE. CAMBRIDGE, MA: MIT PRESS, 1974.

21. SLOVIC, P., VALUE AS A DETERMINER OF SUBJECTIVE PROBABILITY. IEEE TRANS. ON HUMAN FACTORS IN ELECTRONICS, HFE-7, 1966.

22. TVERSKY, A., INTRANSITIVITY OF PREFERENCES. PSYCHOL. REV., 76, 31-48, 1969.

23. TVERSKY, A. & KAHNEMAN, D., BELIEF IN THE LAW OF SMALL NUMBERS. PSYCHOL. BULL., 76, 105-220, 1973A.

24. TVERSKY, A. & KAHNEMAN, D., ANCHORING AND CALIBRATION IN THE ASSESSMENT OF UNCERTAIN QUANTITIES. OREGON RESEARCH INSTITUTE BULLETIN, 1973B.

25. TVERSKY, A. & KAHNEMAN, D., CAUSAL SCHEMATA IN JUDGMENTS UNDER UNCERTAINTY. IN FISHBEIN, M. (ED.), PROGRESS IN SOCIAL PSYCHOLOGY. HILLSDALE, N.J.: ERLBAUM, 1977.

26. VAN DER MEER, H.C., DECISION MAKING: THE INFLUENCE OF PROBABILITY PREFERENCE, VARIANCE PREFERENCE AND EXPECTED VALUE ON STRATEGY IN GAMBLING. ACTA PSYCHOLOGICA, 21, 231-259, 1963.

27. WELFORD, A.T., FUNDAMENTALS OF SKILL. LONDON: METHUEN, 1968.

1. Report No. NASA CP-2180	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AUTOMATED DECISION MAKING AND PROBLEM SOLVING Volume II. Conference Presentations		5. Report Date August 1981
		6. Performing Organization Code 506-54-73-01-00
7. Author(s) Ewald Heer, editor		8. Performing Organization Report No. L-14396
		10. Work Unit No.
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665		11. Contract or Grant No.
		13. Type of Report and Period Covered Conference Publication
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code

15. Supplementary Notes
Ewald Heer: University of Southern California, Los Angeles, California.

16. Abstract

On May 19-21, 1980, NASA Langley Research Center hosted a Conference on Automated Decision Making and Problem Solving. The purpose of the conference was to explore related topics in artificial intelligence, operations research, and control theory and, in particular, to assess existing techniques, determine trends of development, and identify potential for application in automation technology programs at NASA. The first two days consisted of formal presentations by experts in the three disciplines. The third day was a workshop in which the invited speakers and NASA personnel discussed current technology in automation and how NASA can and should interface with the academic community to advance this technology. The conference proceedings are published in two volumes: Volume I is an executive summary; Volume II contains the papers presented at the conference.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Automation, Decision making, Problem solving, Artificial intelligence, Operations research, Control theory	Unclassified - Unlimited Subject Category 66

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	320	A14

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1981

National Aeronautics and
Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

SPECIAL FOURTH CLASS MAIL
BOOK

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

U.S.MAIL

NASA

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return