STRUCTURED ROBOTICS*

M. Weinstein
California Institute of Technology
Pasadena, California

Abstract

Most research in robotics is directed toward the understanding of cognitive processes. But robotics is also an important technological discipline with pragmatic issues to be answered. This paper is addressed to these issues, and particularly to the basic conflict at the core of studies in integrated systems: while the research task is anticipated to be a long and difficult one, it is also expected to yield some practical results in the near future. The paper considers a design and implementation scheme that provides an answer to this conflict. It introduces a structured approach for the implementation of a virtual robot. This approach provides an invariant structure for a system under continuous development, and a method by which the robot can perform useful tasks during its development. The design is currently being applied to the JPL/Caltech robot for planetary exploration, and in parallel to long-term studies, it will provide generations of useful serni-autonomous robots.

Introduction

Most research work in robotics is addressed to one aspect or another of a robotic system (1). However, the design and implementation of a completely integrated system must also consider the global aspects of the system structure and the practical aspects of implementation. The integration of many complex processes, many of which are still not well understood, must fit into an implementation framework that allows smooth integration of completed modules along with the integration of the processes that are still under study.

Early work in robotics recognized the need for such a framework, and some of the problems have been addressed (2, 3, 4). However, later work on integrated systems, and in particular studies performed at SRI, Stanford University, the University of Edinburgh, and in Japan (3, 6, 7, 8, 9), which resulted in a more complete integrated robotic system, was not concerned with the general aspects of robotic design and implementation principles. Perhaps a general disbelief in generality arose during the post-GPS (10) era in artificial intelligence, but it is coming to an end as is evidenced in some recent publications (e.g., 11). But with all the existing records in robotics study, little of the available information on integration is useful, particularly with respect to the design of a robotic system or the principles of its implementation.

In searching for a model and implementation guidelines, the JPL/Caltech work in robotics has also looked for a solution for a basic pragmatic conflict that exists in robotics. While the research task is anticipated to be a long one, it is also expected to yield some useful and practical results in the near future. The JPL/Caltech project, which is concentrating on the study of a robotic system for planetary exploration, attempts to provide various generations of robotic systems that could be used in the near future.

In what follows, the two system aspects of robotics are discussed, i. e., design and implementation. Design must be based on a well structured model for a robotic system that will define all the necessary functions and their relationships. Implementation must be brought under consideration in order to further emphasize the pragmatics of the work. It must impose on the design a strategy that takes into account both priorities and practical aspects. Thus, by combining design and implementation, we arrive at a solution that, on the one hand defines subsystem functions and their dynamic relationships, and on the other takes into consideration the relative difficulty of subsystem development. The combination also guarantees an implementation process with intermediate systems of practical use.

A parallel exists between our approach and that of structured programming (12, 13). A top-down approach with a stepwise refinement — disciplined by the introduction of structure and the use of well defined elements — provides clarity and the ability to verify the correctness of the implementation. But, at the same time, some attractive constructs are sacrificed because the behavior resulting from them is not well defined.

The principles discussed in this report have been used as guidelines for the functional integration of a robot for planetary exploration that is now under study at the Jet Propulsion Laboratory and the California Institute of Technology (14).

Task/Environment Specifications

Robotic systems must not be looked on as general-purpose systems; rather, they are special-purpose systems whose tasks and environment must be well defined. The underlying assumption in robotic systems is that similar tasks can be performed by human beings with or without tools, because the need for a machine to replace a human being implies that the machine is required to perform human functions during its operation. In some cases, this requirement might extend over all functions, but in most instances, the machine will fulfill its purpose by performing only a partial set of the functions required. (This set of functions will be referred to as the functional constraints.) A situation in which a robotic system can perform all functions, that is, complete automation, might be desired but is not necessarily a must.

This work is supported by the Jet Propulsion Laboratory, Contract No. 6146, and by the National Aeronautics and Space Administration, NASA Grant No. NGR-05-002-340.

An example of the functional constraints would be a robotic system for disarming mines in deep seas – the manipulative and sensing functions that must be performed autonomously undersea. All other functions, such as the interpretation of the sensor data or the planning of sequences of manipulations, are optional and can be supplemented interactively by humans. In another example, a chess playing robot may be required to decide on the next move but will not have to move the pieces.

Hence, it is necessary to specify, along with the task/environment, the functional constraints that are essential to the usefulness of the machine.

Functional Specification

The complexity of the environment and tasks will dictate the functional requirements, which, in turn, specify the type of processes the machine must include. Following the definitions and notations of Ref. 15, our discussion is restricted to robotic systems that are based on digital symbol manipulation; we will further assume that the task/environment specifications require a robotic system of the cognitive type, with the general configuration:

<Operative, Cognitive, Conative>

Thus, the operative functions are those that provide interaction with the robot world (e. g., manipulation or sensing). The cognitive functions provide manipulation and interpretation of the work model, such as perception or planning. And finally, the conative functions are the ones that control the goal sequence and specify the robot activities.

In the process of functional analysis, it is necessary to identify all the operative, cognitive, and conative functions that are required; however, it is important not to proceed with the implementation of the processes involved until the designs have been completed and a strategy of implementation has been selected. Thus, it is the designer's task to proceed with functional analysis to a level at which all required processes have been identified and named as functional units; then the modularization of the design and its conceptual verification can be accomplished.

At the level of functional analysis, the conceptual comparison of machine functions to those of humans is essential. It affords a better understanding of the requirements and provides for a scheme by which humans will replace some of the required functions or will complement them. But at the same time, both at the higher level of functional organization and at the lower level of functional process implementation, the robot designer cannot rely on an understanding of the human system, as it has little to offer to process mechanization or to system integration.

Structured Design

With the specification of environment, tasks, and functional requirements completed, structural requirements must be specified to assure a well-structured design. This can be done by determining the requirements of all functional units, and the interactions allowed among them.

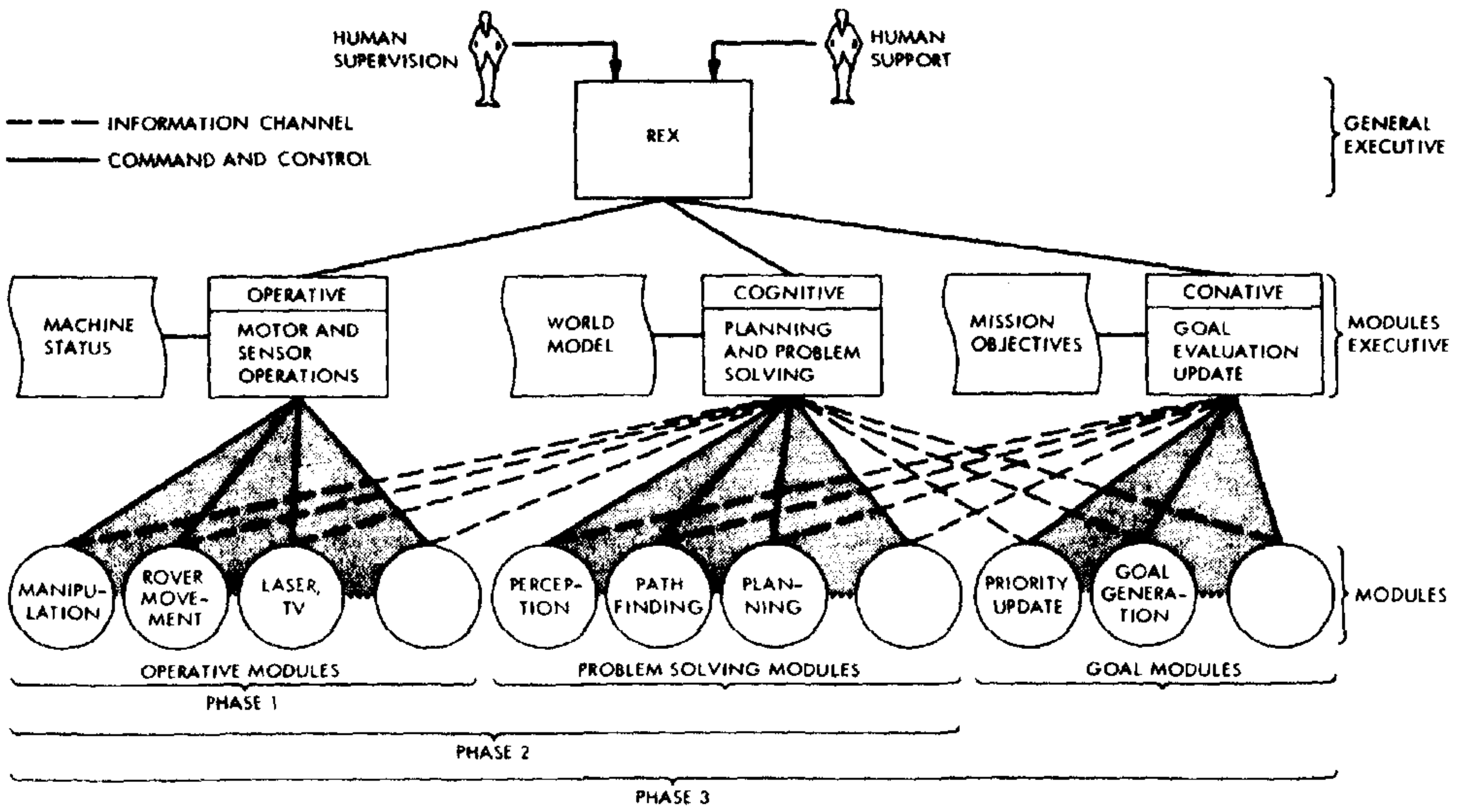
Our structured approach is based on the assumption that the robot design can be completely modularized into well-defined functional units, with a limited interaction that allows all the necessary robotic activities. If such a design existed, however, it might artificially separate units that seem to be closely related and restrict interactions to simple forms that are well understood. But at the same time, it would introduce clarity of functional separation and would ease the robot construction and the verification of its operation.

In this paper, we selected the nonformal approach. The structured design is introduced by the illustration of the Mars Rover design, which is JPL's breadboard for the study of a robot for planetary exploration. The various interactions and their functions are demonstrated through a discussion of the different aspects of the design. The basic functional elements chosen for the design are based on the conceptual model discussed in Ref. 15. A more formal definition of the interactions is given in Ref. 16, which describes the current implementation of the Mars Rover's structure.

The general design is shown in Fig. 1, where the functional modules (the bottom circles) are specifically selected for the Mars Rover. Each of the three major subsystems (operative, cognitive, and conative I consists of an executive element, a data base, and a battery of functional modules. A general robot executive (REX) interfaces among the three subsystems. It is the function of REX to coordinate the operation of the subsystems and to provide the necessary interaction with humans.

The general structure has three levels. On the bottom level, the functional modules are represented by circles. The squares in the second level represent the subsystem's executives. Each subsystem executive is connected by solid lines to the modules under its control, and to a box that represents its data base. Thus, the operative executive monitors the manipulation, rover movement, and sensor modules. Its data base concerns machine parameters that are relevant to operative processes. The broken lines show a direct communication between the modules and other subsystems and indicate that the modules can request information from other subsystems. This operation is also under the control of the modules executive, and it represents one form of interaction between the subsystems. The other interaction takes place through the top and via REX, which represents the third level.

The scheme shows that in such a design for a cognitive robotic system, the robot executive organizes the three subsystems in parallel, while they independently control their own functional modules. While REX provides one kind of subsystem interaction and allows its parallel operation, each of the subsystems independently provides for another subsystem interaction through controlled modules that can again be



Developmental phases of a cognitive robot for planetary exploration

Figure 1

operated in parallel. The control structure maintains some hierarchy that provides status control at all times through REX but, at the same time, allows heterarchical interactions between different system elements.

It is important to note that a functional unit might integrate many processes and keep its own autonomy. However, all interactions with other modules or other system elements must be controlled by the respective executive.

Finally, the functional aspects of the design are described below with respect to the developmental process.

Environment Versus Functional Approach

Once a design has been established, it is possible to begin implementation. However, the difficulties in the implementation of many of the functional modules will make it a long process. Therefore, the building of a robot can take two directions, each representing a different method of simplification. In the first approach, the focus is on the robot world; in the second, it is on the world model. In the first approach, one simplifies the implementation by simplifying the robot world, and in the second approach, the simplification focuses on the world model and its cognitive processes.

The developmental process which corresponds to the first approach is based on the construction of many robotic systems, each suited to operate autonomously in another environment. The environments are simplified to match the state of the art in robotics, and their complexity may grow respectively until it matches some real environment. This approach is in direct contrast to the structured and pragmatic method advocated, and indeed, it has been selected for academic rather than practical reasons, with emphasis on complete autonomy. The robotic work quoted above (5, 6, 7) contains examples of such an approach.

The approach that provides both a smooth development of the robot functions and supports the pragmatic aspects is one which is based on the cognitive aspects. While the robot world is real, its world model and the related mechanized functions are simplified. But to allow the completion of a complex task, an external support must be introduced. Hence, although the actual robot is limited in its autonomous behavior, the virtual robot that relies on external support can still carry out some important tasks.

Therefore, the structured approach must be implemented through developmental steps in which the machine is complemented by external support, and machine autonomy is a tradeoff for pragmatics and structural aspects.

Virtual Robot

The concept of the virtual robot (15) emphasizes two aspects o(the design — an invariant structure and a functional support. These two aspects provide a solution that isolates the basic robotic system structure from the many subsystem changes and developments, and also provides a substitution scheme for functional support at the various developmental phases. Hence, the virtual robot is a complete robotic system that provides for all the necessary functions by the proper substitution of actual mechanized processes by virtual processes. Once the basic structure has been implemented, the robot development will proceed with a gradual replacement of virtual subsystems with others that have been advanced to the state of the art.

The basic principle underlying our solution is the assumption that all robotic system requirements can be met by complementing the mechanized function with human support, particularly the cognitive functions. It is the special characteristic of a robotic system, designed to perform tasks that are regularly performed by humans, that assures us of the existence of solutions and enables us to predict that the virtual robot, which combines man and machine functions, will work.

Those who reject the virtual robot solution as a violation of the complete autonomy that all robotic systems should possess — according to some unwritten ethic — should note that even the human system, which the robot imitates, must follow its own developmental process. A young child is limited in his performance, and for a long time, it is only the ''virtual'' child, which is supported by many external systems, who can survive.

Finally, it should be pointed out that the replacement of a required function in the integrated system does not always require human support. For general studies, testing, and debugging, various methods of simulation might be useful. Simulation schemes will allow not only the study of the simulated functions but, in some cases, the study of other functions that cannot be investigated by themselves. For example, the study of manipulation need not wait for the development of advanced yision systems in order for complex manipulations to be performed in a complex environment. Methods of incremental simulation (17) for the vision function can provide object descriptors that are useful both for manipulation studies and as a guideline for vision requirements.

Robot Implementation

There are three states of robot implementation that are important to our approach:

(1) Prefunctional
(2) Functional
(3) Completion

The first state represents a robotic system where the functional consttaints have not yet been met. For example, for a robot designed for Moon exploration, all the operative functions must be implemented before it will reach the functional state. Hence, the robot must be able to manipulate objects, to look around, and to collect and send sensor data — all autonomously. But once these functions, which are specified by the functional constraints, have been implemented, the system is in the functional state. In this state, the robot can perform all its tasks virtually. The degree of autonomy depends on the developmental state of the virtual robot. Finally, the completion state indicates complete autonomy.

The subject of interest to our discussion is the functional. This state does not represent a single phase of development but rather the whole developmental stage, where the robotic system is integrated as a virtual robot and is going through a gradual mechanization of its virtual functions. The developmental process will be discussed and illustrated using our schemes for the developmental phases of the Mars Rover.

The first diagram (Fig, 1) shows the final structure of the robot. The functional state will be achieved by, first, the mechanization of the operative subsystem and, second, the completion of the integrating structure. The integrating structure will provide a mechanism by which all necessary cognitive and conative functions will be available to the system through external support. It is usually the function of REX to provide for the support when the necessary subsystem is missing. But when all subsystem executives have been implemented, it is the respective executive which organizes its own functions and supplements them.

Figures 2, 3, and 4 show several phases in which the robot is in its functional state as a virtual robot; each diagram shows only the mechanized elements. The three phases and their subphases presented are:

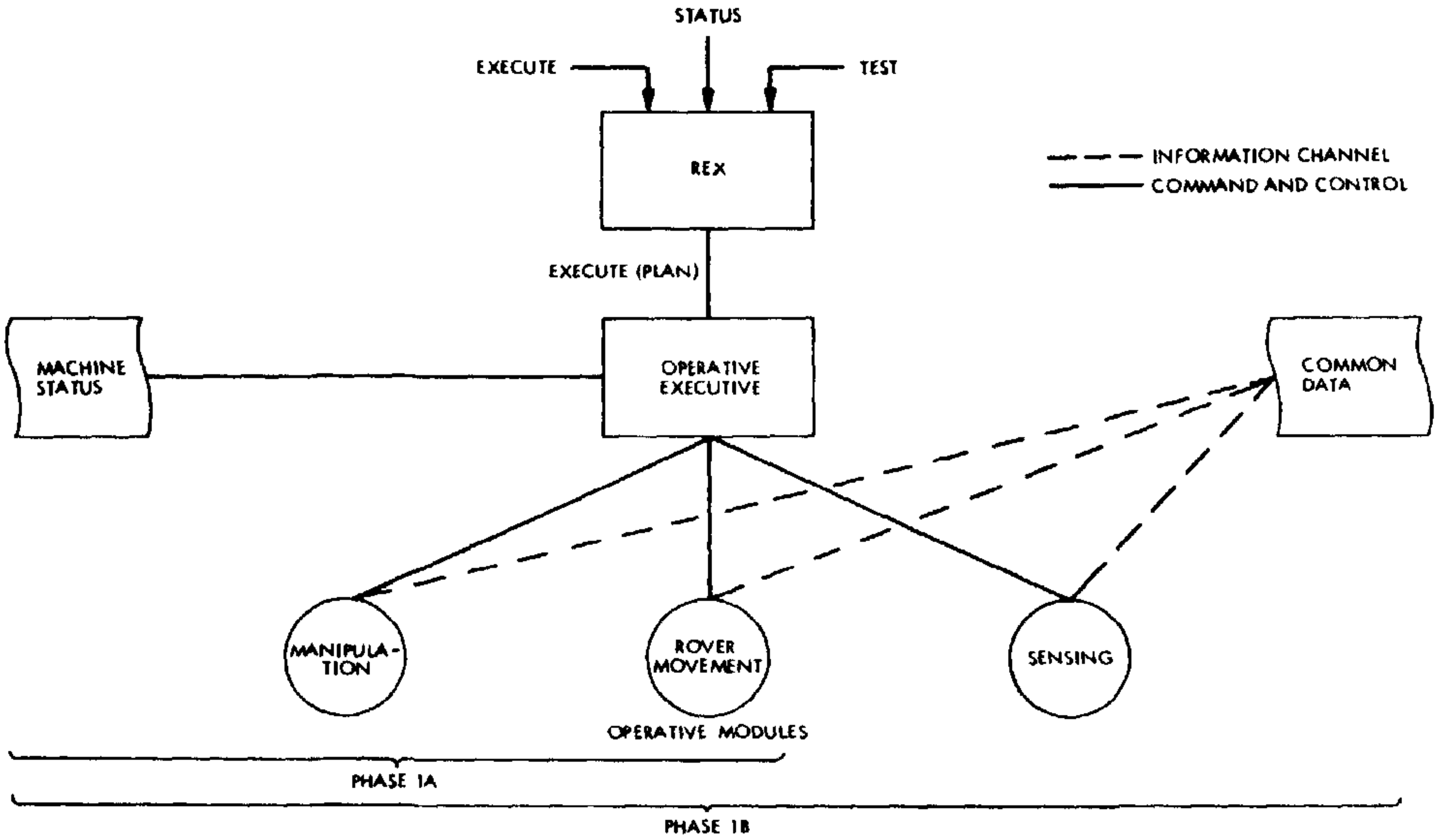
Phase 1: Operative
Phase 2: Cognitive
Phase 3: Conative

In Phase 1, the Mars Rover is expected to have complete control of all motor and sensor functions. This includes mobility, manipulations, and the ability to collect sensor data from dual-TV, laser, and proximity sensors (the current sensor systems under study).

In Phase 2, the system will be upgraded by the essential cognitive functions. It will include scene analysis for the detection of objects and obstacles in natural environments, pathfinding and navigation modules, and a planner.

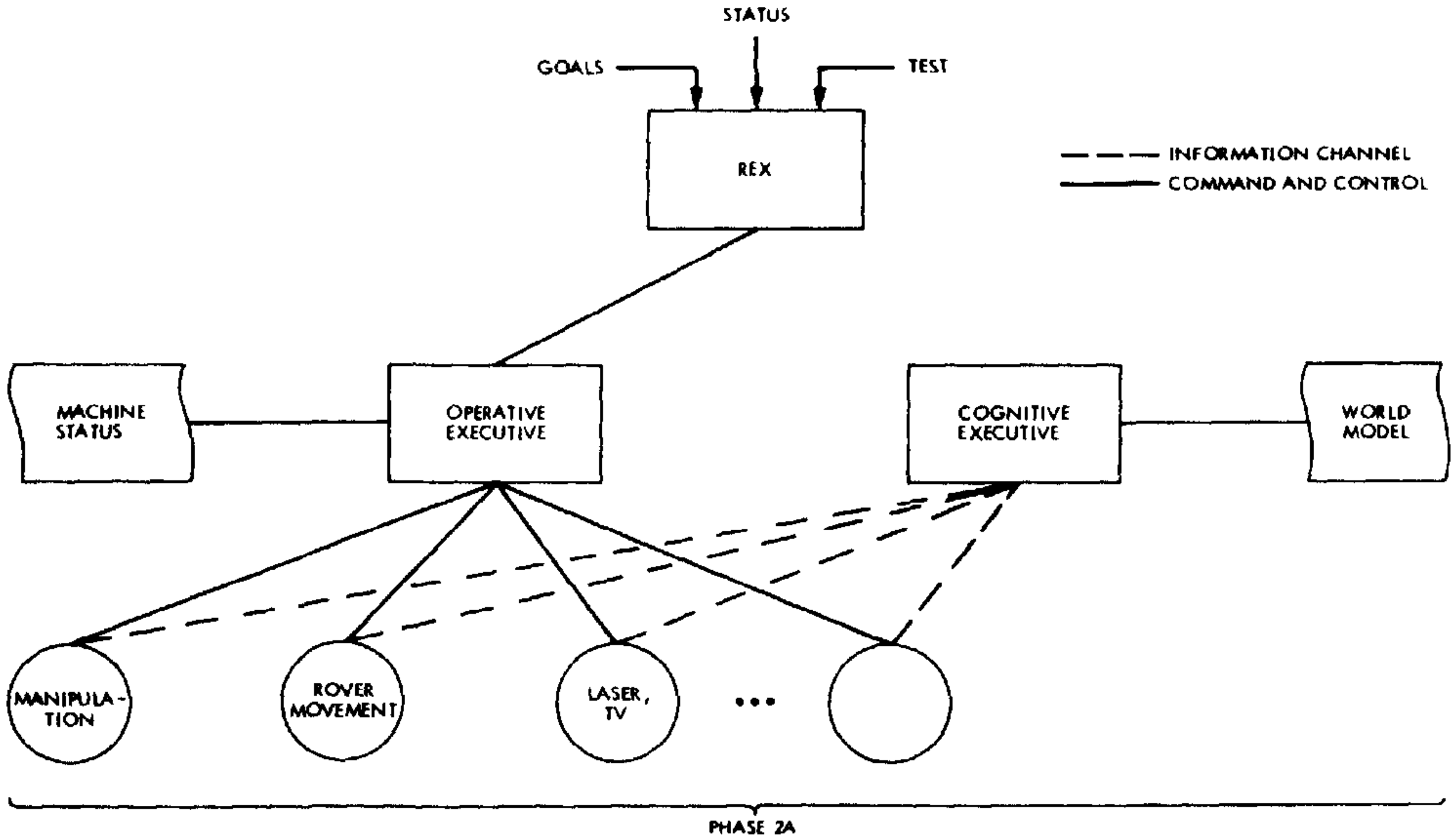
Finally, in Phase 3, the system will contain the processes for environment-driven goal generation.

The diagrams illustrate a scheme for functional development of the Mars Rover or of any other cognitive robotic system with similar operative function constraints. Figure 2 shows Phases la and lb. In these phases, a complete specified action plan is sent via REX to the operative, which then monitors its execution by the robot operative modules (three of which are shown). Along with this execution mode, REX provides for tracing the robot activity, status control, and testing mode. All the information needed for the plan's execution is given in the data base called "machine status." However, in an advanced operation, it is expected that the modules will require additional common information; thus, in Phase lb, an additional data base is shown.



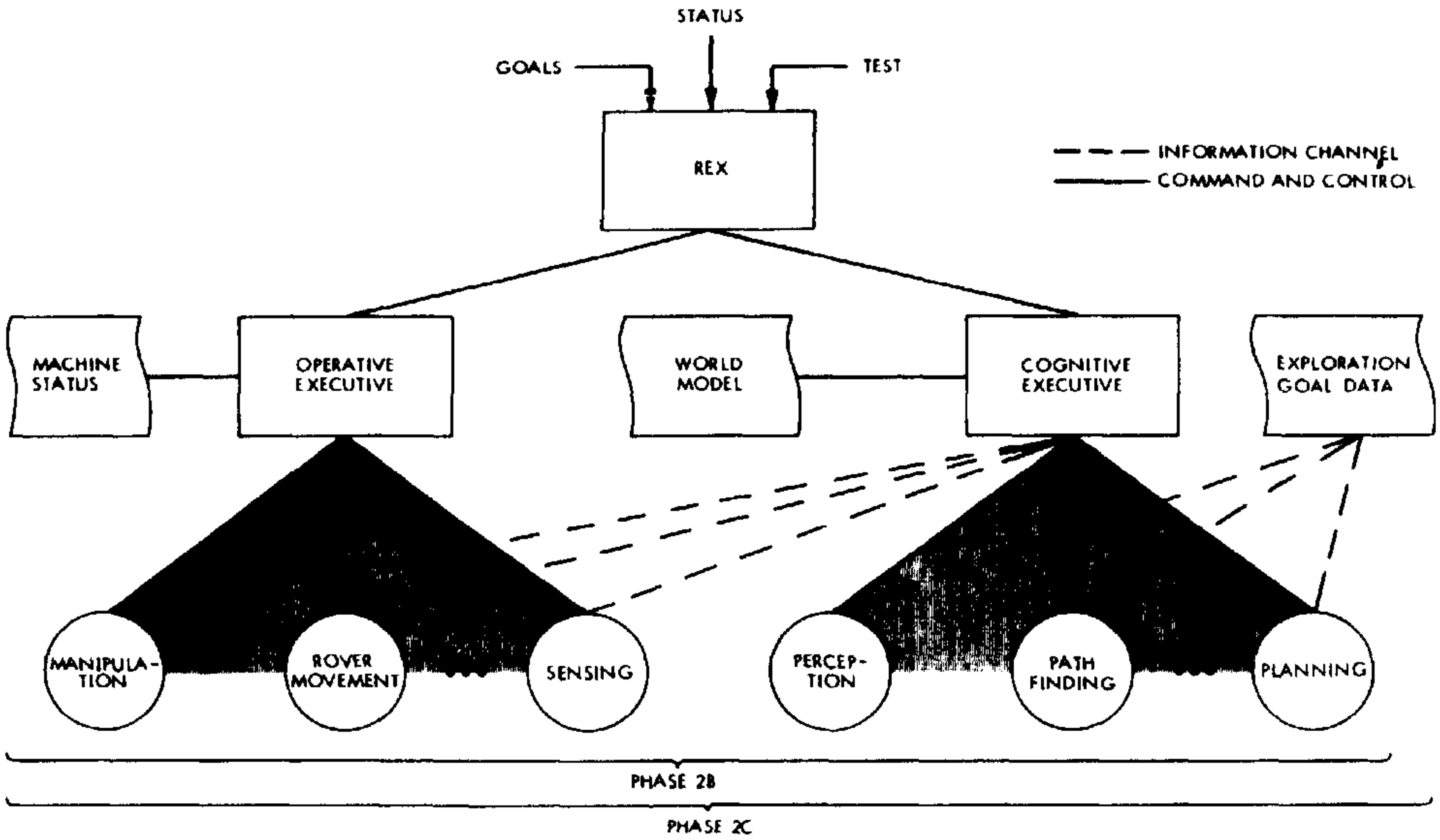
Basic rover control

Figure 2



Extensive rover control (current implementation)

Figure 3



Cognitive-supported rover

Figure 4

The extension of the common data is shown in Fig. 3, which describes an early state of Phase 2. When the common data items grow, they become the initial world model that is used to describe the robot world. To manage the growing data base and to control its support to the operative subsystem modules, another executive system is added. This marks the emergence of the cognitive subsystem. The initial function of this subsystem is to manage the information flow in and out of the world model. In later phases, such as 2b (Fig. 4), the cognitive subsystem will be extended and connected to REX to provide more functional support. The extension shown is achieved by supplementing the cognitive with its own functional modules, that is, with cognitive processes that provide *the* extension *of* the world model and the improvement of its use.

Some of the functions of the cognitive subsystem can be illustrated by its developmental stages in the following example. When a description of a free path between two points is needed, the information is requested from the cognitive executive. The information might be readily available, in which case the cognitive task is a simple one. If, however, it is available indirectly in a map description, the cognitive executive must call a special pathfinding module to process the data and to provide the free path description. Finally, if no data are available, it might be necessary to activate a plan that directs the proper sensor to collect the data, and then to activate a scene analysis process to yield a map with obstacle descriptions.

Hence, the management of the world model can be extensive and might initiate a chain of activities that include noncognitive functions. The important aspect of this activity is that the cognitive subsystem is a centralized knowledge base that serves all other system elements. It is the sole responsibility of the cognitive subsystem to make all global decisions. When, for example, the manipulator needs some essential information, the system control structure does not allow it to proceed on its own and to get the information but forces the module to direct its request to the cognitive executive, which is the only system element equipped with the necessary tools for the proper decisions.

In conclusion, we have shown three levels *of* activities that are controlled by the cognitive system and represent three levels of development: first, the management of existing data and world model descriptors, second, the extension of the passive descriptors by dynamic processes that can either generate the description or serve as procedural descriptors by themselves, and third, the planning level, where a plan must be formed and activated to achieve a given goal. Although some requests of the operative modules might result in planning, most of the planning activity takes place in response to requests directed from REX.

It is during this planning activity that another type of information is needed, i. e., the evaluation of various robotic states that might be considered as subgoals. This information is concerned with the goals of the robot mission. In our functional structure, we provided a special data base that is concerned with such information, and it is the goal structure on which the conative subsystem operates. In the example of the Mara Rover or other planetary exploration robots, each mission will have its own goal structure, where the mission goals and their relative priorities are specified. It is important to note that a change in a mission can be achieved through a change in the goal structure alone, leaving the world model untouched. It was this aspect of the goal structure that suggested the separation of the conative functions from the cognitive.

In the Apollo missions for Moon exploration, the function of goal evaluation and selection was carried out by a human team on Earth, and it might be expected that the same will be done with respect to a robot system. But phase 2c illustrates the first step in extending the robot autonomy in this direction. Finally, in Phase 3 (Fig. 1), it is expected that, with the development of the conative system, the evaluation of robot states and the dynamic generation of new goals will be mechanized. This last mechanization will allow extensive autonomy for the robotic system for long periods of time.

References

1. Winston, P. H., "The MIT Robot," Machine Intelligence 7, B. Meltzer and D. Michie, eds., Edinburgh University Press, Edinburgh, Scotland, 1972, pp. 431-463.

2. Nilsson, N. J., and Raphael, B., "Preliminary Design of an Intelligent Robot," Computer and Information Sciences, Vol. 2, Academic Press, Incorporated, New York, New York. 1967, pp. 235-259.

3. Raphael, B., "Programming a Robot," IFIPS Congress Proceedings, Edinburgh, Scotland, August 1968.

4. Rosen, C. A., "Machines that Act Intelligently," Science Journal, October 1968, pp. 109- 114.

5. Raphael, B., Robot Research at Stanford Research Institute, A.I. Center, Technical Note Number 64, Stanford Research Institute, 1972.

6. Feldman, J. A., et al.. "The Stanford Hand-Eye Project," Proc. of the IJCAI-69. Association for Computing Machinery, May 1969. pp. 52L526a.

7. Barrow, H. G., and Crawford, G. F., "The Mark 1.5 Edinburgh Robot Facility," Machine Intelligence 7, B. Meltzer and D. Michie, eds., Edinburgh University Press, Edinburgh, Scotland, 1972, pp. 465-480.

8. Ejiri, M., et al., "An Intelligent Robot with Cognition and Decision-Making Ability," Pruc. of the IJCAI-71. British Computer Society, 1971, pp. 350-358.

9. Kinoshita, G., et al., "Pattern Recognition by an Artificial Tactile Sense," Proc. of the IJCAI-71, British Computer Society, 1971, pp. 376-384.

10. Newell, A., Shaw, J. C., and Simon, H. A., "Report on a General Problem-Solving Program for a Computer," Information Processing: Proc. International Conference Information Processing, UNESCO, pp. 256-264. Also printed in Computers and Automation, July 1959.

11. Minsky, M., A Framework for Representation of Knowledge, MIT: AIM Number 306, 1974.

12. Dahl, O.-J, Dijkstra, E. W., and Hoare, C. A. R., Structured Programming, Academic Press, London, England, and New York, New York, 1972.

13. Knuth, D. E., "Structured Programming With Go To Statements," ACM Computing Surveys, Vol. 6, No. 4, December 1974, p. 261.

14. Lewis, R. A., and Bejczy, A. K., "Planning Considerations for a Roving Robot with Arm," Proc. of the IJCAI-73, August 1973, pp. 308-316.

15. Weinstein, M., A Framework for Robotic Design, California Institute of Technology. To be published in the Proceedings of the Second USA-Japan Computer Conference, Tokyo, Japan, August 1975.

16. Weinstein, M.. and Stevens, K., Robot Design: The Operative Subsystem, Caltech Information Science Technical Report No. 16, June 1975.

17. Woods, W. A., and Makhol, J., "Mechanical Inference Problems in Continuous Speech Understanding," Proc. of the IJCAI-73, August 1973, pp. 200-207.