N90-25561

BEHAVIORAL NETWORKS AS A MODEL FOR INTELLIGENT AGENTS

Nancy E. Sliwa
MS 152D
NASA Langley Research Center
Hampton, Virginia 23665-5225

ABSTRACT

This paper describes on-going work at NASA Langley Research Center in the development and demonstration of a paradigm called behavioral networks as an architecture for intelligent agents.

This work focuses on the need to identify a methodology for smoothly integrating the characteristics of low-level robotic behavior, including actuation and sensing, with intelligent activities such as planning, scheduling, and learning. This work assumes that all these needs can be met within a single methodology, and attempts to formalize this methodology in a connectionist architecture called behavioral networks.

Behavioral networks are networks of task processes arranged in a task decomposition hierarchy. These processes are connected by both command/feedback data flow, and by the forward and reverse propagation of weights which measure the dynamic utility of actions and beliefs.

An experimental prototype of a behavioral network testbed is being developed in the Intelligent Systems Research Lab. This work is augmented by grants with Old Dominion University and the University of Maryland.

JUSTIFICATION

As NASA's mission repertoire continues to favor large, complex, long-duration missions, design and operations costs and manpower commitments could come to dominate NASA's budget and activities. This would limit NASA's ability to start new programs, hampering NASA's quest to continue expanding the frontiers of knowledge, understanding, and technology. Operational activities must be made less resource-demanding, more efficient. Increased assistance from computers and intelligent systems is one possible means of maintaining future flexibility in operational commitments.

NASA is currently on the threshold of operational deployment of its first-generation artificial intelligence systems. It appears that at the current state of the practice, the best payback to NASA is in the development of relatively small, single-purpose expert systems. These systems are appearing in launch processing, mission control[1], Shuttle-based experiments, and are baselined for elements of Space Station Freedom operation.

However, NASA's future mission plans call for elaborate, complex, and interconnected systems that integrate not only different functionality, but which span the multiple spectra of symbolic and numeric computation, human and robotic activity, and high and low speed and bandwidth requirements. The class of tasks to be performed by such systems involve handling perception, cognition, action, and reaction with smooth simultaneity. The ideal system would also modify its behavior appropriately based on feedback and its history of performance, and be relatively easy to develop. Unless these future requirements are addressed today, the capability will not be available tomorrow.

Myriad research projects exist that ably address specific components of these needs, such as planning, resource allocation, and learning. This isolationist approach assumes that, after all problems are "solved" independently, the solution techniques can be stirred together into a complete system. It can be argued that unless all requirements are considered together, this resulting composite system will fail to integrate those requirements in a satisfactory way.

A fundamental assumption of this research is that there exists a single methodology that smoothly blends all requirements into a single architecture. This research is concerned with developing such a methodology that provides a "seamless fit" among the broad spectrum of activities and abilities of an intelligent system, including planning, scheduling, resource allocation, execution control, perception, and learning. Systems developed under such a methodology would provide NASA with the type of intelligent systems required for future missions. This methodology should allow modules of different functionality to be developed using similar techniques and to work together smoothly. For example, consider a lunar outpost for LOX production. A power allocation system, a crew activity planning system, and the control systems of surface robots could potentially all be developed in the same framework, and would be able to act and interact intelligently.

This paper provides a general motivation and description of the behavioral network concept, and discusses some issues associated with this approach. Current work is summarized.

GENERAL MOTIVATION

There seem to be two general classes of approaches to intelligent system research. One is the development of a "bag of tricks," an accumulation of techniques that are applied as suitable to a particular problem type. This approach has worked well for the area of computer programming, and is typical of young, immature areas of technology. The second approach is the development of a general theory and methodology applicable across most if not all of the problems in the field. This approach is usually more successful with a mature, well-understood technology. However, it is important, even in young technology areas, to continually examine successful "tricks" and attempt to formulate unifying approaches. This research project falls in this latter class.

The original work in the Intelligent Systems Research Lab in intelligent task decomposition and control focused on hierarchical levels of activity. A system that connected a blocksworld procedural planner to a jointed manipulator with an end effector and simple sensors was developed[2]. Experience with this development revealed several desirable attributes of a methodology for intelligent agent development:

1. A methodology must be able to connect symbolic and numeric programming approaches.

2. A methodology must be able to connect slow processes and fast processes.

3. The environment and the goals of an intelligent agent change dynamically, both from its own actions and from changes in the external environment. A methodology must unite goal-driven planning and reactive planning in a cooperative way.

4. A methodology must be able to blend control from both the intelligent agent itself and a human operator.

5 Sensor-closed control loops are very effective. A methodology should incorporate them.

6. The concept of hierarchical levels is relative; a function can be "higher level" or "lower level" than another, but architectures which defines precise levels are forcing arbitrary cuts in a continuum for the sake of convenience.

7. The same is true of the concept of heterarchy. A function has more or less interaction with other functions. Heterarchical architectures make arbitrary cuts in this continuum for convenience.

8. Actuators and sensors can be treated isomorphically. Actuators have a sensory component (proprioception), and sensors have an actuation component (positioning and activation).

9. Most robotic system development efforts never consider a general solution to the resource allocation problem. Most such systems are very resource constrained, and use customized solutions to the problems of redundant resources or resource failure. A methodology should provide a resource allocation technique that handles these problems and provides maximum parallelization of activities though appropriate resource allocation.

DESCRIPTION OF BEHAVIORAL NETWORKS

Based on these observations, the concept of behavioral networks was developed[3]. Behavioral nets represent a hybrid among classical control techniques, artificial intelligence planning techniques, and connectionist approaches.

Fundamentally, a behavioral network can be thought of as an acyclic directed graph whose nodes represent specific functions, or behaviors, of an intelligent system, with two-way links which propagate information including functional parameters and weights. The net flows from top to bottom in the task decomposition sense. That is, a node is linked downward, to "children" nodes, if the accomplishment of the child node's function is required to accomplish the original node's function. Put another way, a node accomplishes its goal or function by instantiating subgoals in the form of children behavior nodes.

Functions

Each node is built from a "template" (figure 1), which can be represented as a classic feedback control loop, receiving input as to the desired state X_d from a parent node, and receiving feedback Y concerning the current state or setting from children nodes. In the general case, the node would be required to compute the current state X_c from Y, i.e. $X_c = f(Y)$, where Y is a vector of the feedback signals. The node function then computes the required command parameter vector C to the children nodes that minimize the difference between the desired and current states, (i.e.: $X_d - X_c \rightarrow 0$), passes C to the children nodes, and passes X_c upward to its parent nodes in turn. In classical planning research, this equates to the selection and instantiation of an operator schema to minimize the "distance" between current and desired states via means-ends analysis.

This functional aspect of behavioral nets is similar to the concepts of structured programming approaches in both intelligent planning research and in control theory. However, behavioral nets provide a continuous flow of control in an isomorphic structure from potentially high-level, symbolic behavior, to low-level numeric control functions, providing a way to smoothly integrate goal-oriented behavior and reactivity. In addition, given the

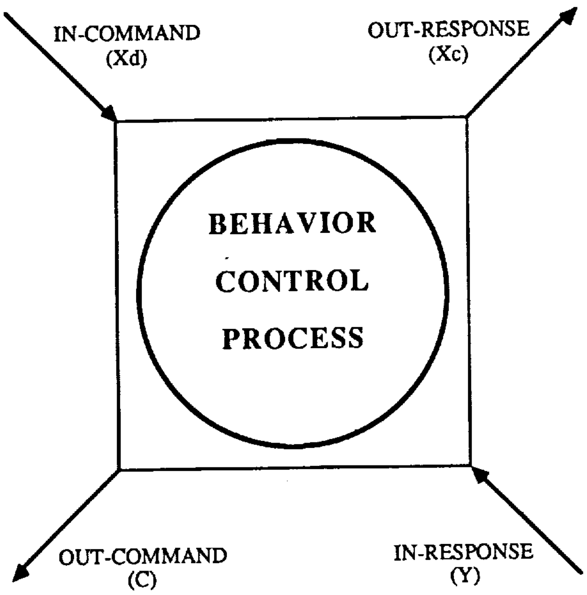

Figure 1. - Behavioral Network Node Template

command and feedback components of the node template, each behavior has some degree of both motor and sensory functionality, providing a way to isomorphically represent actuation and sensory behaviors.

Weights

Another aspect of behavioral nets concerns the propagation of "weights" or potentiation level/threshold measures. This gives behavioral nets their connectionist flavor. Commands (goals) from parent to child are weighted according to utility measures, including the probability of success and the need or urgency or the action. Feedback from child to parent is also weighted according to similar parameters from a reactive or sensory point of view.

These weights are combined within each node, at each execution cycle, and form updated weights for use in the next cycle. Thus weights are propagated both "upward" and "downward" in the network, as are the command parameters and feedback. A threshold switching function requires each node's weight to exceed an established threshold before the node can "fire." This threshold varies with the utility cost of an action.

In general, a node will have more than one parent, and multiple children (figures 2 and 3). The choice of which child to activate when, and with what parameters, is determined by the combinations of weights at each cycle of the network. This provides a method for choosing among competing subgoals, and for sequencing subgoals.

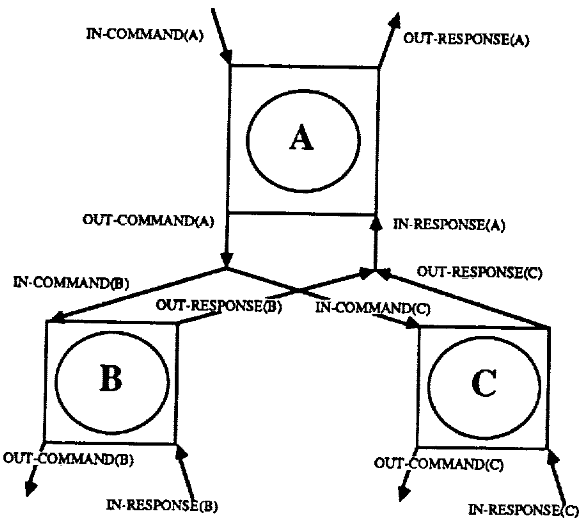

Figure 2. - Parent Node with Multiple Children

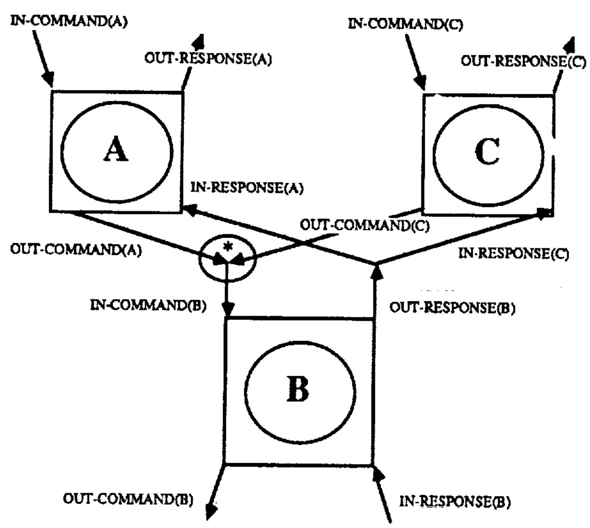

Figure 3. - Child Node with Multiple Parents

The weighting scheme represents a real-valued logic ranging between -1 and 1, with 1 representing total belief/desire, -1 representing total disbelief/avoidance, and 0 representing ignorance/indifference.

The weights serve several purposes. They determine when a behavior has been activated, and arbitrate among goal and resource conflicts. They provide a way to blend control from multiple sources, including from a human operator. They also alter the network over time according to the feedback, either increasing or decreasing the probability that a behavior is activated or that it successfully competes with other behaviors.

Parallelism

Parallel activity is an inherent part of behavioral networks. In the behavior net theory, each node is assumed to be continuously active if its threshold is exceeded, and is continuously checking its inputs and weight status. Thus the network is continuously adapting to new goals, and to changes in the environment. This also provides a technique for dealing with processes of varying speeds. Behaviors work at their own speeds asynchronously, using the most recently available command and feedback data from their parents and children.

True parallelism can of course only be achieved in a multiprocessor environment. The ability of a behavioral network-based system to maintain its operational integrity in a time-shared environment would require careful design and static resource allocation.

RESEARCH ISSUES

Behavioral approaches are being investigated in other research groups. Brooks' work[4] in subsumption architectures is behavior-oriented, but does not attempt to establish a methodology. Brooks' goal is to build systems which "do the right thing;" the resulting architectures have highly convoluted wiring and logic, are not easily duplicated or understood, and are not extensible. Our research in behavioral networks, on the other hand, attempts to duplicate the functionality of Brooks' systems within a structured, easily developed, and extensible architecture.

Behavioral approaches have also been used quasi-operationally. The Hughes Corporation used a behavioral approach in the Autonomous Land Vehicle work for DARPA[5]. This system chose optimal path components from weighted alternatives. That approach is representative of most behaviorist research to date: techniques for choosing among alternatives, or disjunction. In addition to providing a structured approach to behavioral system development, this research is also attempting to expand the abilities of such systems to handle conjunctive activities, both sequentially and in parallel. Several pertinent issues relating to these extensions are discussed below.

Conjunctive Behaviors

To discuss the issues involved with sequential behaviors, consider three activities, A, B, and C. Most connectionist systems attempt to accumulate evidence in favor of a choice among the alternatives: A or B or C. Most tasks, however, involve being able to sequence behavior: A and then B and then C. For example, to pour a glass of soda, get a glass, and then open the can of soda, and then pour the soda in the glass. If multiple resources are available, some tasks can be done in parallel: e.g., A while B, and then C. For example, if two people are available, one can get the glass while the other opens the can of soda. Both of these activities must be completed, however, before the soda can be poured into the glass. A and B are preconditions of C.

Behavioral network theory is attempting to determine a method for structuring the network and propagating weights that allow maximum parallelism while appropriately sequencing activities, as well as choosing among alternatives. Sequences and alternatives should emerge naturally as a result of the dynamic variation of the weights within the network, instead of being per force programmed into the code procedurally.

Goal Maintenance/Achievement

Goals and subgoals in an intelligent system are generally classed as goals of achievement or goals of maintenance. Goals of achievement are those that must be achieved at some point, but to which the system is indifferent thereafter. Goals of maintenance must be achieved and maintained for a period of time to establish the preconditions of later actions. Using our example above, the glass must continue to be available and the can must continue to be open until the action of pouring the soda into the glass is complete. Thus A and B must be maintained until C is achieved. The current research is working to establish a means of making this distinction intrinsically within the network. Precondition goals of maintenance are inherently resources that are required for subsequent actions. Weights propagated through the network should be able to maintain this logical resource availability, just as it obtains and maintains physical resources for tasks.

Network Structuring

The approach to decomposing a problem into subtasks has heretofore been very ad hoc. Many decompositions of one problem are possible, and no heuristics exist to rate one decomposition against another. Therefore, an effort is being made to formalize the decomposition task itself, in an attempt to optimize and eventually automate the decomposition process.

Lattice structures have long been known to offer a decomposition of partially ordered sets which exhibit algebraic structure. Dr. David Livingston at Old Dominion University developed a method to generate task decompositions by constructing the lattice of substitution property (SP) partitions on a state machine model of the task. Given the SP lattice, a group of partitions that will yield a "good" decomposition are selected[6]. Given the decomposition network resulting from this process, fourth-order constraint satisfaction networks have been able to find a path from the initial state to the goal state, thus generating a plan to perform the task.

This decomposition approach provided a method for finding decompositions, but has proved to be computationally intractable, and still reliant to some degree on heuristics. However, initial work by Dr. Livingston indicates that self-organizing networks could be used to find good, though non-optimal, decompositions in constant time. Continuing work under this grant is investigating more fully this approach, and integrating the task decomposition and planning techniques within the self-organizing network paradigm.

Connectionist Planning

A grant with Dr. James Reggia of the University of Maryland is currently investigating the application of connectionist competitive activation techniques to planning, scheduling, and resource allocation. Work to date has concentrated on searching the literature for connectionist approaches to similar problems, and on executing different problem solutions on the University of Maryland MIRRORS/II connectionist simulator. Three problem domains and functions have been successfully developed: satellite camera resource cooperation to maximize target coverage[7], Voyager resource sequencing during planetary flybys, and fault interpretation and recovery for satellites[8]. These experiments demonstrate the ability of connectionist models to handle different function types, but more stringent problem models are necessary to determine their usefulness for complex problems.

Prototype System Development

A prototype implementation of a behavioral net system is being developed on a Symbolics 3620 in Lisp. This prototype, called Behavioral Network Functionally Integrated Testbed (BeNeFIT), will provide a means of testing and analyzing the performance of behavioral networks on a range of problems. This prototype will be used primarily to demonstrate the mechanics of the behavioral network structure, and to investigate various methods for propagating weights thoughout the network.

CONCLUSION

A methodology is in development to provide a structured and extensible approach to the design and development of behavior-driven intelligent agents. Behavioral networks are task decomposition networks which propagate commands, data, and feedback in a structured programming sense, and which propagate weights in a connectionist sense. Key research issues hinge on the ability of the network to represent the task in a logical way, to combine sequential, parallel, and alternative behavior in a single structure, and to handle the distinctions between subgoal achievement and maintenance. Supporting work at the University of Maryland and Old Dominion University is in progress. A prototype testbed implementation is being developed at NASA Langley Research Center for demonstration and research purposes.

REFERENCES

1. Muratore, John, "Application of Artificial Intelligence to Space Shuttle Mission Control," AAAI Conference on Innovative Applications of Artificial Intelligence, Stanford, California, March 1989.

2. Orlando, Nancy, "A System for Intelligent Teleoperation Research," AIAA Computers in Aerospace IV Conference, Hartford, Connecticut, October 1983.

3. Sliwa, Nancy and Donald Soloway, "A Lattice Controller for Telerobotic Systems," 1987 American Controls Conference, Minneapolis, Minnesota, June 1987.

4. Brooks, Rodney, "A Robust Layered Control System for a Mobile Robot," IEEE Journal of Robotics and Automation, RA-2(1), April 1986.

5. Daily, M., Harris, J., et al., "Autonomous Cross-Country Navigation with the ALV," Proceedings of the DARPA Knowledge-Based Planning Workshop, Austin, Texas, December 1987.

6. Serpen, G. and D. Livingston, "A High-Order Boltzmann Machine Configuration for Transfer Sequence Searches in Decomposed State Machines," presented at the International Joint Conference on Neural Networks, Washington, D.C., June 1989.

7. Bourret, P., Goodall, S., and M. Samuelides, "Optimal Scheduling by Competitive Activation: Application to the Satellite Antennas Scheduling Problem," Proceedings of the International Joint Conference on Neural Networks, Washington, D.C., June 1989.

8. Bourret, P. and J. Reggia, "A Fast Interactive Satellite Failure Diagnosis Procedure," Proceedings of the Goddard Conference on Space Applications of Artificial Intelligence, Greenbelt, Maryland, 1989.