A SITUATED PLANNING AGENT FOR THE VIRTUAL WORLD ENVIRONMENT

by

DAVID CLAYTON CROUCH

(Under the direction of Donald Nute)

ABSTRACT

The following paper describes an agent designed for the Virtual World environment, based upon the principles of situated cognition. Situated or embedded cognition is an idea in opposition to the Physical Symbol System Hypothesis as to how intelligent agents are to be constructed. Agents based on these principles employ task-based decomposition rather than function based, and do not rely on explicit symbolic representation, centralized representation schemes, or centralized control. This project aims to demonstrate the fitness of the situated approach in a problem domain designed for symbolic agents, and to demonstrate that situated agents are not limited to mere reactive behavior by presenting a situated agent capable of plan construction and execution, as well as mapping and navigating behaviors.

INDEX WORDS: Situated, Embedded, Planning, Navigation, Physical Grounding Hypothesis, V-World, Distributed Representation, Distributed Control System.

A SITUATED PLANNING AGENT FOR THE VIRTUAL WORLD ENVIRONMENT

by

DAVID CLAYTON CROUCH

A.B., The University of Georgia, 1998

A Thesis Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2002

A SITUATED PLANNING AGENT

FOR THE VIRTUAL WORLD ENVIRONMENT

by

DAVID CLAYTON CROUCH

Approved:

Major Professor:	Donald Nute
Committee:	Ron McClendon Walter D. Potter

TABLE OF CONTENTS

CHAPTER 1: A NEW APPROACH TO INTELLIGENCE

Throughout much of the history of Artificial Intelligence, efforts to produce autonomous agents capable of interacting with dynamic and complex environments have been based on metaphors derived from digital computer architectures. This approach has resulted in the tendency to divide intelligence along functional boundaries. The separation of input, output, data storage and processing in computers has manifested in artificial agent design as barriers of abstraction between sensing, acting, knowledge representation, and control systems. The dominance of this model of intelligence is apparent in the structure of the discipline itself. Many of the sub-fields of Artificial Intelligence were created to solve pieces of the puzzle as dictated by the digital computer metaphor. Knowledge Representation, Computer Vision, Computational Intelligence, Logic Programming, and Natural Language Processing, among others, all seek to provide working parts of an agent possessing the capabilities of a human intellect. The assumption underlying this effort, that these parts when combined will produce a functioning entity, is left largely unspoken.

Newell and Simon (1976) provide an explicit formulation of this model of intelligence in the Physical Symbol System Hypothesis. Briefly, this formulation describes intelligence as the operation of a reasoning system on sets of arbitrary symbols. This model implies that data relevant to the agent are represented as symbols, which are then manipulated according to a set of rules to produce other sets of symbols, which are then converted into actions. The influence of the digital computer metaphor is evident in the separation of sensing, representing, reasoning, and acting. However, the assumption that the separately designed modules will produce human-level intelligence when linked together continues to be made without supporting evidence from experimentation or from structural knowledge of natural intelligence. There is nothing to suggest that such boundaries of abstraction exist in that which is being modeled, or that such a modular system has any hope of duplicating the desired effects. In addition to theoretical problems, this approach has also resulted in several practical difficulties that will be discussed later in this work.

Some scientists in the field are beginning to look at problems encountered in Artificial Intelligence research as inherent in the approach, rather than in the subject matter. New lines of inquiry are being suggested that discard assumptions of modularity and explicit symbolic representation. Researchers at MIT have offered the Physical Grounding Hypothesis, a model arising from rejection of these and other assumptions, as an alternative to the Physical Symbol System Hypothesis (Brooks, 1990). This model states that the flexible and adaptive behavior considered indicative of intelligence emerges from the manner in which inputs, or sensors, are connected to outputs, or actuators. Representation and reasoning are held to be implicit properties of the interactions between inputs and outputs, rather than separate functional components of the system. Inspired by insights from the examination of biological intelligence and by analysis of the failures of symbol-system-based agents, this model rejects explicit symbolization and explicit reasoning on the grounds that they are unnecessary and detrimental to the effort to produce agents capable of intelligent behavior.

The agent presented in this work is intended to be a demonstration of the Physical Grounding Hypothesis. The Virtual World domain, for which the agent was designed, is a development environment for agents employing planning, reasoning, and learning strategies based on the Physical Symbol System Hypothesis. The agent's performance in this domain will serve to demonstrate both the agent's effectiveness in achieving the goals of the domain, and the particular benefits of agents based on the Physical Grounding Hypothesis. The rest of this chapter will examine the Physical Symbol System Hypothesis and the Physical Grounding Hypothesis in greater depth, exploring the strengths and weaknesses of both and examining how the latter arose as a solution to some of the problems that plague the former.

1.1 THE PHYSICAL SYMBOL SYSTEM HYPOTHESIS

In order to understand the motivation for the Physical Grounding Hypothesis, it is important to examine the paradigm from which it departs. The following sections examine the history and particulars of the Physical Symbol System Hypothesis in order to make apparent the need for a new approach to the duplication of intelligent behavior.

1.1.1 Inspiration

The Physical Symbol System Hypothesis was proposed by Newell and Simon in order to offer a computer scientist's perspective on the nature of intelligence and to suggest an approach for duplicating it with computers. This hypothesis makes strong claims about what an intelligent system, whether natural or artificial, must consist of, and equally strong claims about the path that must be taken by the field of computer science in order to eventually create artificial systems with the cognitive power of human intelligence. The Physical Symbol System Hypothesis was inspired by the observation that other areas of science have been revolutionized by the discovery of laws of qualitative structure. The germ theory of disease, the doctrine of atomism in chemistry, the theory of plate tectonics in geology, and the cell doctrine in biology are all examples of paradigms that provided quantitative units by which one can explain observations and draw meaningful conclusions about the subject matter. One can identify and treat the disease by identifying the germ, understand the terrain in terms of the tectonic plates that gave it form, know the properties of the solution by knowing the properties of the ingredients, and understand the functioning of the organism by the composition and interactions of the cells from which it is built. Newell and Simon sought to provide a similar quantitative principle from which to understand intelligent systems. Accordingly, they developed the concept of the physical symbol system, a system realizable on physical architectures that contains some number of symbols that can represent any expression and a set of processes to create, modify, reproduce, or destroy these symbols. The symbols themselves are arbitrary physical patterns whose meanings can be designated and interpreted by the rules of the system. Any system having these properties "...has the necessary and sufficient means for general intelligent action" (Newell and Simon, 1976).

1.1.2 Support

The evidence for the necessity and sufficiency of the Physical Symbol System Hypothesis is empirical rather than theoretical. At the time that this hypothesis was formalized, the viewpoints it encompasses had already been in use for several decades in the fields of Artificial Intelligence and Cognitive Science. Numerous small victories had been won by researchers in both fields. Artificial systems that were able to solve specific problems, such as playing chess and constructing simple plans, with some degree of intelligence were taken as evidence toward the sufficiency of the hypothesis. Symbol-system based models of human cognition that were able to duplicate certain human behaviors, such as solving simple problems and understanding natural language utterances in limited domains, were taken as evidence toward the necessity of the hypothesis. Just as finding the specific germs responsible for specific diseases led to a general theory of germs and diseases, proponents of the Physical Symbol System Hypothesis hoped that successful handling of specific problems would lead one day to a general knowledge of how intelligent systems operate. It is worth noting that this justification offers the Physical Symbol System Hypothesis as an interim model of intelligence, and assumes that it will be eventually replaced by a model that produces not just specific solutions to specific problems, but a general understanding of the functioning of intelligence. The formulators of this model of intelligence were also encouraged by the lack of any viable competing model (Newell and Simon, 1976).

1.1.3 Ramifications

The Physical Symbol System Hypothesis makes a number of strong claims about the nature of human intelligence and the effort to produce artificial intelligence. According to this hypothesis, any system intended to display a level of intelligence anywhere near that of human beings must be designed as a physical symbol system, and any natural system that displays such a level of intelligence must have a physical symbol system as a basis for behavior and decision-making. This implies that any generally intelligent system will have a physical symbol system as the basis for its reasoning. It also implies that physical symbol systems are capable in and of themselves of producing the kind of general-purpose problem solving behavior exhibited by human beings, given the proper set of operators for creating, destroying, duplicating, and manipulating symbols. The search for such a set of operators, or for principles dictating what characteristics such a set of operators must have, has been the preoccupation of the majority of researchers in Artificial Intelligence since its emergence as a cohesive sub-field of Computer Science. While this endeavor has resulted in advancements in the handling of specific problems, as well as minor advances in the search for general purpose decision making algorithms, there does not seem to be definite progress in the direction of systems with capabilities similar to human beings.

The Physical Symbol System Hypothesis also implies that human beings themselves reason using operations on arbitrary physical symbols that represent objects or expressions. Structural knowledge of the human brain available at the time may not have been able to confirm or deny this assertion. Twenty-six years of advancement in psychology and neuroscience, however, have yet to demonstrate the presence of discrete individual entities that might correspond to the symbols described by the hypothesis (Brooks, 1991). While the brain does employ electrical signals in its operation, there is no conclusive evidence that these signals designate objects or properties in the sense implied by the Physical Symbol System Hypothesis. That the brain represents reality and the objects in it in some fashion is apparent from the effectiveness of the decisions and actions that the brain generates, but there is no indication of explicit representation on the level of description suggested by Newell and Simon. There is also no indication of a separation in the human brain between representations and operators that create, destroy or modify them. Even if we oversimplify brain function by describing two patterns of activation as two separate thoughts, we must observe that the only characteristics that make them distinct are the signal strength and the paths taken through the brain. Both of these characteristics are determined by the existent structure of the brain, which in turn is determined and modified by the signals. The signals themselves have no individual identity aside from how they are shaped by the medium. Thus, there are no explicit operators that manipulate symbols in the brain, indicating that the hardware-software distinction borrowed from digital computers cannot be usefully applied to the brain above a certain level of description. While this alone does not invalidate the Physical Symbol System Hypothesis or explain its failure to duplicate human-level intelligence, it does suggest that a model of cognition that takes the structure of biological computing architectures into account may be more successful at fulfilling this goal.

1.1.4 Practical Difficulties

Researchers in Artificial Intelligence working within the Physical Symbol System Hypothesis have encountered a number of barriers to progress toward the eventual goal of a generalized intelligent system. One of the implications of the Physical Symbol System Hypothesis is the necessity of heuristic search as a component of generalized intelligence. The concept of heuristic search is here applied as generating and modifying symbol structure so as to produce a symbolization of the solution to a problem. Since the Physical Symbol System Hypothesis holds that it is necessarily constituent of any intelligent system, all systems must generate intelligent behavior through the operation of heuristic search on symbol systems. Unfortunately, there are several problems that arise when one tries to apply the notion of heuristic search to such a broad goal as the creation of generally intelligent systems.

The dependence of heuristic search on knowledge about the solution space presents a problem for artificial systems, particularly when the solution space is extremely complex or broad in scope. The effectiveness of heuristic search, as opposed to simpler methods such as brute-force exhaustive search, derives from information about the search space that allows the search routine to focus its efforts in directions that seem more productive. While this is not an insoluble problem in specific domains that are well understood, it presents difficulties when dealing with complex and dynamic environments. The knowledge needed to direct a heuristic search is not always available, and it is not always readily apparent what knowledge is needed. We do not always know what information we ourselves use in certain types of problem solving, since much of the reasoning process takes place without full conscious awareness. The difficulty of identifying the type of information needed, and of making that information available to the heuristic search routine, imposes limits on its effectiveness as a tool for creating generalized intelligence. The fact that heuristic searches frequently require much more knowledge of their domains than most biological intelligences are equipped with to produce the same level of performance is also a concern, from the viewpoint of duplicating the effectiveness of natural systems.

Effective heuristic search also requires a meaningful symbolization of the solution space in order to be effective. Whatever the problem at hand, it must be presented to the search routine in such a way that a series of incremental steps connects the beginning state with the solution state. The solution space must be presented such that the operators available to the routine can identify the best direction at each increment of the search process. This requires considerable analysis on the part of the designers before the problem is given to the search routine, even in the case of specific, well understood domains. To produce an artificial system whose capability for intelligent action spans across even similar domains would require a symbolizing component with as much or more intelligence than one should expect from the program as a whole. If the symbolization is done by human designers, then the end result cannot be properly considered a generally intelligent artificial system. It would be merely a specialized tool for a given domain, an aid to human intelligence rather than an intelligent system in its own right. Any reference to a trans-domain symbolizing routine that may or may not be developed in the future is merely postponing the problem. Heuristic search cannot be a necessary or sufficient component of generalized intelligence if external assistance is required to prepare the problem space. The goal is to eventually match the capabilities of human intellects, and that requires the system to handle a broad range of problem domains as they are.

Heuristic searches run into considerable difficulties when faced with dynamic environments. The effectiveness of a heuristic search hinges on a suitable symbolization of the solution space. In any but the simplest of domains, rendering the beginning state, the solution, and the intervening search space as a cohesive and organized set of symbols requires some effort on the part of the program itself or the designers of the program. When the problem-solving environment changes overtime, this task increases in difficulty proportional to the rate and extent of the changes in the environment. The more work that must be done to interpret the environment to the heuristic search routine, the more trouble the routine is going to have keeping up with changes in the environment. This problem has caused researchers to look forward to the development of computer vision systems and other artificial modes of input capable of swiftly symbolizing the problem environment. Such ideal systems, however, have not appeared, and show no signs of doing so any time soon. The difficulty faced by such systems, as observed in the paragraph above, lies in the interpretation that must be performed on raw data to make it useful to the search routine. Relying upon input routines that maintain up-to-the-minute symbolic representations of the domain does not seem to be an option, and without such assistance heuristic searches cannot hope to function effectively in environments that change over time.

1.1.5 Methodological Difficulties

There is also a potential flaw lurking in the very heart of the methodology underlying the Physical Symbol System Hypothesis. The inspiration for the importance of symbols in this model derives from the development of logic, which offers a means separating the process of reasoning from the subject matter. It was this facet of logic that led Newell and Simon to formulate the Physical Symbol System model as a system of operators manipulating sets of symbols that could stand for any object or expression (Newell and Simon, 1976). The division between the structure and content of reasoning allows for the rigorous analysis of methods and strategies for manipulating expressions before these methods and strategies are implemented. The use of formal logic provides a degree of certainty that is otherwise not available. To those working within the Physical Symbol System Hypothesis, it seems only natural to attempt to impart that clarity and precision of thought to artificial reasoning systems. It is here that a methodological mistake is being made. Observation of biological intelligence, the phenomenon that Artificial Intelligence aims at modeling and understanding, does not suggest that certainty is a design priority. On the contrary, natural intelligent systems seem prone to sacrifice certainty in favor of quickness of action. From human beings down to the simplest microscopic organism, nature seems to prefer a "good enough" decision-making strategy to one that makes perfect choices. Perfect choices take time and resources, and natural selection has apparently found brevity superior to certainty. While the prevalence of sub-optimal strategies of natural intelligence does not eliminate the possibility of perfectly rational intelligent systems being designed, it may suggest that imperfect reasoning is good enough for, and possibly essential to, the sort of broad-scope effective decision-making that is the goal of Artificial Intelligence. Duplicating the capabilities of systems that exist seems a much more sensible mission than attempting to build systems that improve on phenomena that we are only beginning to understand.

Logic can be described as a formalization of the subset of human cognitive strategies that most consistently produce solutions that turn out to be accurate. While an artificial system built to reason with capabilities similar to those of humans may reasonably be expected to have the capacity to use logic, it is impractical at the current stage of exploration and development to require our creations to perform better than we do. Logic is a grand achievement resulting from the endeavor to advance human reasoning and avoid the pitfalls to which human cognition, left to itself, is prone. However, what capability we have for rational thought arises from the same cognitive infrastructure that gives us the fallibility that logic was created to prevent. We cannot hope to duplicate the full scope of the intellectual power of our species by ignoring that which does not meet the high standards that we as scientists set for ourselves. The majority of Artificial Intelligence researchers are guilty of attempting to design not systems that think as we do, but systems that think as we feel we should. This error of intent, while understandable, must be corrected if the field of Artificial Intelligence is to achieve its goals of understanding human cognition and improving the reasoning capabilities of computing devices. Hidden assumptions and subconscious motivations must be identified and clarified if we are to avoid becoming trapped in an endless search for an ideal to the detriment of our goal.

1.1.6 Conclusion

To the extent that one views the Physical Symbol System Hypothesis as an interim explanation and a research program, the support given for it is sufficient. However, the success of this hypothesis at providing solutions to small problems in specific domains does not justify its permanent acceptance as dogma for the field of Artificial Intelligence. A theory whose application allows researchers to build systems capable of duplicating specific isolated effects is not a general theory of intelligence. Likewise, attempting to improve upon the performance of the object of study should not precede successful duplication of it. The Physical Symbol System Hypothesis was intended to guide research to discover a more general theory of the functioning of intelligence. The paradigm has experienced successes and failures, both of which have increased our understanding of intelligence and made us more aware of the remaining gaps in our understanding. The following section presents a discussion of the Physical Grounding Hypothesis, an alternative paradigm that may, given some exploration and development, prove to be successful in guiding the attempt to produce artificial systems with the intelligence of human beings.

1.2 THE PHYSICAL GROUNDING HYPOTHESIS

The Physical Grounding Hypothesis offers an alternative approach to the duplication of general-purpose intelligence, inspired in part by the methodological and empirical failures of the Physical Symbol System Hypothesis. Advocates of the Physical Grounding Hypothesis, also known by other names such as situated cognition or embedded cognition, note that the concept of intelligence need not be restricted in scope to include only the most advanced tasks of which humans are capable. Tasks such as playing chess, generating proofs in first-order logic, and communicating via human languages represent the highest degree of intelligence available for observation, but this does not mean that simpler tasks do not involve some level of intelligence worthy of study. The duplication of intelligence using computers may be more feasible if we approach it one step at a time, instead of attempting to duplicate the most complicated behaviors first. The Physical Grounding Hypothesis implies that a broader concept of intelligence may be advantageous to the attempt to understand it and to reproduce its effects.

The central idea of this hypothesis is that, in order for a system to be intelligent, it must employ representations that are grounded in the real world. This idea contrasts with the architecture of most systems designed using the Physical Symbol System Hypothesis, in which the representations take the form of a set of symbols representing an ontology of objects and properties. By using the world as its own model, the agent avoids the need for complicated representation schemes and the imposition of artificial categories when dealing with the external environment. The Physical Grounding Hypothesis discards the notion that intelligent agents must have a single representation and a single control system, as well as the assumption that the two functions must be separate. The metaphor of the digital computer is done away with, reducing such concepts as representation and control system to terms of convenience rather than structural guidelines. Functional decomposition, in which sensing, acting, representation and control are handled by separate modules, is replaced by behavioral decomposition, in which each behavior is handled by a separate subsystem, and the functions formerly made explicit by the Physical Symbol System Hypothesis emerge as the result of the way in which sensors are connected to actuators (Brooks, 91).

The Physical Grounding Hypothesis aims to remedy several assumptions, techniques, and habits of thought that have dominated much of the research and ideas in Artificial Intelligence to its detriment. The Physical Symbol System Hypothesis offered a way to understand the way intelligence works by comparing it to the workings of a digital computer. This viewpoint has since become the way to view intelligence, to the extent that research on brain function has been influenced by the tendency to look for structures corresponding to familiar computer components. While the assumptions upon which the Physical Symbol System Hypothesis is based were educated guesses that deserved to be explored, the limitations that the approach has encountered signal the need for a change of paradigm. The next few sections elaborate on the design methodology of the Physical Grounding Hypothesis by describing how this approach handles problems that the Physical Symbol System Hypothesis has encountered.

1.2.1 Implicit vs. Explicit Symbolization

One of the barriers to progress within the Physical Symbol System Hypothesis stems from reliance upon explicit symbolization. The desire to build intelligent machines based upon the proven techniques of logic is understandable, given the origin of Artificial Intelligence in the field of Computer Science, which in turn has its roots in mathematics. While building an intelligent system based on explicit symbols and operations on symbols does ensure that its inner workings will be intelligible to observers, this is at the very least not a necessary component of intelligent action (Brooks, 1991). At worst, insisting upon this intelligibility can add difficulty to an already difficult task. The use of identifiable representational units acted upon by a separate and domain-independent set of operators requires computationally expensive interpretation to be performed at every step. All input and output must be converted to the language of arbitrary symbols, so that the various functional modules of a symbol-system based agent can communicate. This added computational burden reduces the speed and overall reasoning capacity of the system. While an internally intelligible system would be beneficial from the perspective of understanding the process of intelligence and debugging the systems while they are being constructed, such an arrangement still sacrifices processing power for the sake of an entirely separate goal from what the system is designed to achieve. An artificially intelligent system may not be something that one can plug a monitor and keyboard into.

Systems designed according to the Physical Grounding Hypothesis avoid the cost of internal symbolization by avoiding it entirely. Since this viewpoint identifies intelligence as a property of the interaction between an agent and its environment, rather than as a property of the agent's internal workings, explicit representation of inputs and outputs becomes unnecessary. The connections between the agent's sensors and actuators can be designed such that the strength of a signal from a sensor, or the rate of activation, or how it is connected in order to excite or inhibit other sensors, actuators, and pathways in between can convey all of the information necessary for the agent to choose an appropriate action quickly. In behavior-level decomposition, each set of connections manipulates the flow of information in a manner specialized to a given behavior. This eliminates the need for a common language between components or for a domain-independent central decision-making process.

1.2.2 Distributed Data Handling vs. Categorization

Explicit symbolization creates further problems that are ontological, rather than practical, in nature. Symbolic logic, an important component of the physical symbol system described by Newell and Simon, is a language of objects and properties, and of truth or falsity. While this language is typically how humans speak and think consciously about the world, it is not the same manner in which we interact with the world. The first kind of knowledge is propositional, containing discrete categories. The second kind, which implicitly underlies the first and allows it to be adequate for communication, is operational knowledge, which manifests primarily in behavior and takes into account that everything is a matter of degree. There is no easy way to put into words how one maintains one's balance, or judges distance, or decides whether something is round. An ontological system based on discrete categories cannot easily represent a world with no sharp boundaries. What defines an edge? Does an apple with a slice cut out of it still constitute an apple? In order to handle such a world, a symbol system must draw artificial lines of abstraction, which inevitably obscures some information about the world. A symbol system that draws more lines to capture the in-between states may be more perceptive, but it will be slower as a result. A symbol system that works quickly by drawing less lines and creating more general categories will lose more information contained in the grey areas of reality.

A further problem with the categorization required by a symbol system-based reasoner is that there is more to be known about the world than simply what is in it. Much of the information that is useful to agents interacting with the real world concerns relationships among objects in the world, or between objects in the world and the agent itself. A banana peel on the floor to your left is qualitatively different from a banana peel under your foot. The problems with this sort of knowledge are that it is difficult to express concisely, and difficult to express completely (Brooks, 1990). Instructing a system never to drop a lighted match does not tell it what happens if it allows the match to touch something flammable while holding it. To impart that information by giving a system causal knowledge of objects in the world is a task of such overwhelming difficulty that no one has attempted it on a general scale. Designing systems that infer any portion of the full complexity of the world is similarly difficult, in part because we the designers cannot call to mind everything we know, nor explain how we came to know it. A symbol system by definition must carve up the world into categories, and information is always lost in the process.

Physically grounded systems circumvent this difficulty by exchanging centralized control and representation for distributed control and representation. Categories are tools of communication, not tools of action. The fact that each behavior in a physically grounded system implicitly includes control structures and representational schemes specialized to that behavior eliminates the need for the system to have general knowledge of the world, or to organize that knowledge into hierarchies for easier handling. While consistency of action may imply some operational knowledge of categories, such as in the case of a robot that cleans up empty soda cans (Brooks, 1990), grounding representation in the physical world avoids the need for making categories explicit. Since no behavior need interact with another behavior other than through excitation or inhibition, there is no need for any behavior module to speak any language but its own, or to render its knowledge of the world propositional. This strategy allows different modules to employ different priorities in observing and behaving, so that no relevant information has to be sacrificed in the name of centralized representation.

1.2.3 Uninterpreted vs. Interpreted Visual Data

Another troublesome assumption implicit in the Physical Symbol System Hypothesis is the very possibility of general-purpose symbolizing of information about the agent's environment. Symbol-system based reasoners, in order to produce meaningful outputs, require symbolization of inputs that captures the data relevant to the task at hand. In any process of representation, there are trade-offs to be made. In vision systems, representing color with a high degree of accuracy means less resources to devote to the representation of texture, or orientation in space, or any other perspective. While this problem does not arise as much in the handling of specialized domains at which symbol system-based programs excel, it does present a barrier to the creation of general-purpose intelligence within this paradigm. There is too much data in an environment as complex as the real world to symbolize all of it, with its multitude of objects, characteristics, and relationships. Choices must be made about what to include and what to leave out, and everything that is left out decreases the agent's scope of ability. Richness of detail, variety of purpose, and keeping track of changes on a regular basis must all be taken into account. A system that, in addition to these three priorities, must also balance the time and resources needed to translate its output into the language spoken by other components of the system cannot represent reality with the speed and richness necessary for general-purpose intelligence (Brooks 1991).

The behavior-level decomposition of agents based on the Physical Grounding Hypothesis rescues this methodology from some of the difficulties of general-purpose computer vision. Since every behavior in the repertoire of a physically grounded agent is a separate subsystem, the vision system does not have to decide what to do with the information or how to interpret it. Each separate subsystem can take whatever information is relevant to it from the sensors, with little or no manipulation required. Thus the system avoids the time investment, resource usage, and general difficulty of converting input from a data-rich environment into a representation language that works for a variety of purposes. With behavior-level decomposition, the data can be passed to the subsystems unrefined, to be implicitly interpreted as each subsystem handles the data in a manner specialized to the behavior it governs. While symbol system-based vision becomes more and more ponderous as the representation scheme is scaled up to satisfy a greater variety of behaviors, the same increase in generality in a physically grounded system requires only that new behavioral connections be properly integrated with existing connections. Scaling this methodology up to the level of human functionality will not be an easy task, but as a research direction it does seem capable of circumventing the practical barriers that the Physical Symbol System Hypothesis has run up against.

1.3 A SITUATED AGENT IN A SYMBOLIC ENVIRONMENT

In practice, Physical Grounding Hypothesis-based, or situated, agents tend to display greater flexibility, greater responsiveness, and more effective handling of uncertainty, noise, and change in the environment than their symbol system-based counterparts. What is not immediately apparent at first glance, however, is how an architecture based on distributed, direct connections between sensors and actuators can be anything other than reactive. Appropriately, a significant amount of research has gone into answering that question since the inception of the Physical Grounding Hypothesis roughly twelve years ago. Meeting the challenge of building agents capable of executing plans and keeping track of location, while retaining the benefits granted by tight coupling between sensing and acting, is an important step in determining the limits of this new paradigm. The agent presented in this paper is intended to solve problems that require planning and mapping, both of which require non-reactive longer-term decision-making strategies, while deviating as little as possible from the principles of modular behaviors connecting sensing to acting and using the agent's environment as its own representation.

The Virtual World Environment, in which the agent operates, was designed to aid in studying and creating agents that plan, learn, and navigate based on techniques derived from work within the paradigm of the Physical Symbol System Hypothesis. Situating the agent in this arena allows for a direct qualitative comparison between its performance and that of agents using symbol system-based strategies. The purpose of the agent presented here is to succeed where symbolic agents have succeeded, while demonstrating both the practical benefits of environmental grounding of representation and a greater degree of resemblance to the behavior patterns of biological agents. This demonstration will hopefully show that the Physical Grounding Hypothesis has the potential to greatly advance the field of Artificial Intelligence toward the goal of duplicating general-purpose human level intelligence, thereby understanding the inner workings of intelligence and increasing the capabilities of computers.

The next chapter discusses the Virtual World environment, as a development environment and a testing ground for navigating agents, planning agents, and learning agents, as well as the structure of several symbol system-based planning agents that have successfully completed the tasks presented in the planning environment. The third chapter will discuss the implementation of the agent presented here, and the fourth will discuss the results of that implementation and other possible applications for the principles whose validity the agent is intended to illustrate.

CHAPTER 2: THE VIRTUAL WORLD DEVELOPMENT ENVIRONMENT

2.1 PURPOSE

Virtual World, or V-World, is a simulated environment into which artificial rational agents can be placed in order to observe and test their capabilities. As a development environment, this program is suited to the design of agents using insights from the Physical Symbol System Hypothesis. The Physical Symbol System Hypothesis assumes that the component modules of intelligence implied by a functional decomposition approach can be built and debugged separately, then connected to create an intelligent agent. The V-World environment provides researchers with an arena in which to develop the component responsible for tasks related to reasoning, planning, and learning.

Researchers following the Physical Symbol System Hypothesis tend to see the problem of creating human-level intelligence as one that can be solved in large functional increments (Brooks, 1997). First design, debug, and perfect the component modules of intelligence, such as perception, reasoning, and representation, then connect the modules and test in a static and artificial environment free of noise, so that the interfaces between the modules can be debugged. After this is done, subject the system to dynamic and noisy environments for further testing and debugging. The idea was to avoid dealing with the issues of noise and complexity inherent in a dynamic environment by focusing first upon development of the modules themselves and the interfaces between them. It was thought that handling change in the environment and lack of crisp perception and action would be easier once a fully functional intelligent agent was available that was capable of navigating simplified domains. To this end, a number of testing environments have been built that allow the problem of intelligent action to be solved one component at a time.

The V-World Program provides an environment in which researchers can focus on developing strategies for building the reasoning component of intelligent agency. Information about the surroundings of a V-World agent is given to the agent in pre-symbolized form. This saves the designer from having to create an artificial vision system that translates the environment into a meaningful and useful symbolic language, a problem that still has yet to be solved. The agent's actions in its environment also occur in symbolic form. The agent program delivers one of a small set of symbols to the V-World program itself, without the need for building separate systems responsible for translating the agent's decisions into behavior in the environment. This arrangement allows the researcher to focus on the problem of determining which behavior is best suited to a given set of environmental circumstances or global goals. The pre-symbolized format of inputs and outputs isolates the problem of reasoning from other problems such as perception and effective locomotion, so that the development of effective planning and learning strategies need not wait for breakthroughs in other sub-fields.

V-World provides a means to create artificial environments for the purpose of developing techniques for learning, planning, and navigating. The individual world environments can be set up to provide challenges in any of these domains. Learning worlds present agents with unfamiliar phenomena with which the agent must determine how to interact. Planning worlds provide the agent with a global goal that can only be accomplished through the orderly execution of a series of sub-goals, which requires that the agent be capable of formulating, carrying out, and sometimes modifying plans. Navigation, which partakes somewhat of planning and learning, is required for the survival of the agent in any world, as the agent must locate useful or necessary resources and find paths to them as needed. The variety of possible agent environments enables the study and development of artificial reasoning techniques that can later be combined with the other components of intelligence to produce functioning agents possessing human-level intelligence.

2.2 STRUCTURE

The V-World program creates a two-dimensional grid of squares, in which the agent and other actors can move and interact. V-World generates a series of discreet spaces having pre-specified geographical configurations based on the configuration of the current world, which can be designed by the user to create environments tailored to the problem one wishes to model. The survival or continued execution of the agent program is dependent on two factors, strength and damage. Strength is expended during each move by the agent, whether that move consists of changing position, pushing on something, or sitting still. Strength can be replenished by various means, which are specified in the world file. Death results if the agent's strength reaches zero. Damage is accumulated by contact with actors or situations that are defined as harmful in the world file. The amount of damage which is fatal to the agent and means by which to reduce accumulated damage are also specified in the world file.

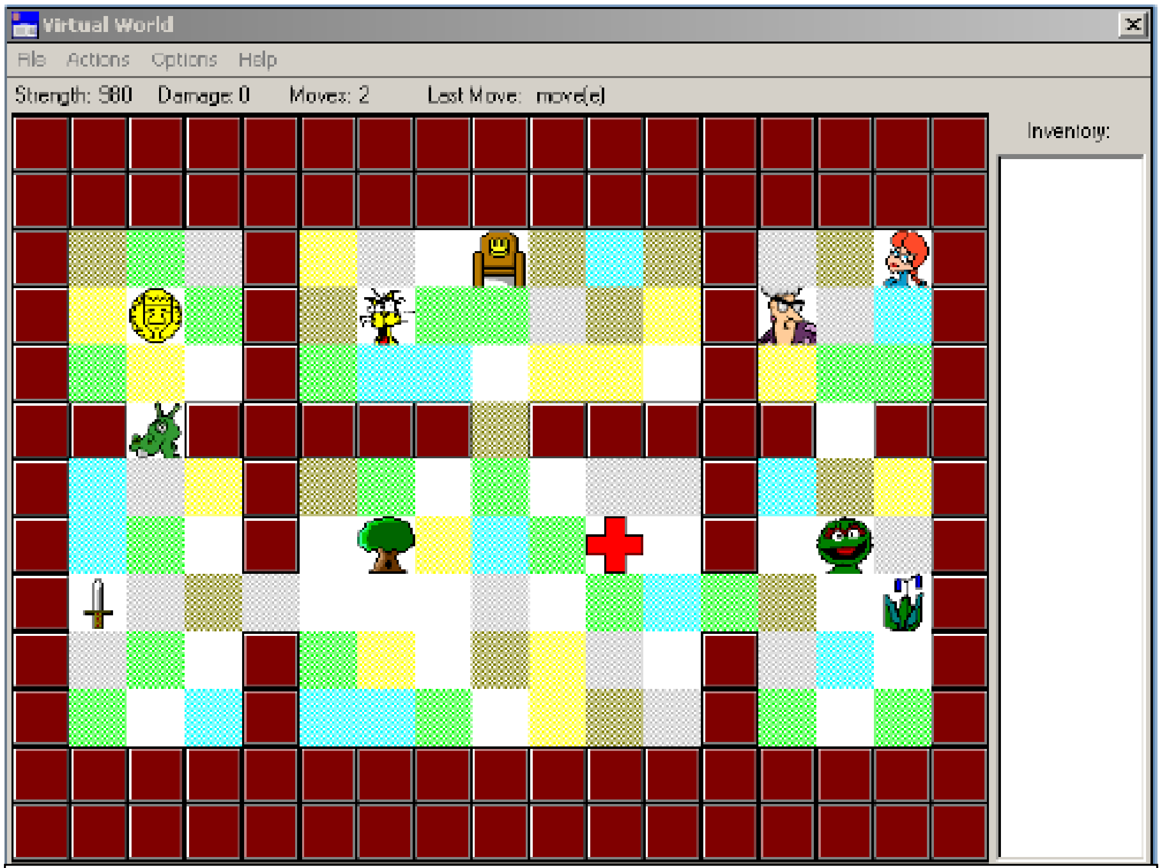


Figure 1: The V-World Planning Domain

2.2.1 The Worlds

V-World provides the developer with a variety of options for constructing problem spaces. The squares in the grid can be occupied by several different types of content. The more fundamental types include empty space, which allow the presence and passage of agents or actors, and walls, which do not. The squares may also be occupied by trees and crosses, which can be assigned properties, such as healing or replenishment of strength, that characterize their interaction with the agent. The squares may contain collectible objects, which alter the agent's capabilities, and can be added to the agent's inventory by occupying the square. Finally, a given square may contain an animate actor, which may be harmful or beneficial to the agent, depending on what parameters are specified for the world.

These varied possibilities allow worlds to be constructed with a variety of goals in mind for the agent. For example, a world in which the agent's task is survival, necessitating strategies for locating and remembering sources of strength and healing, might contain a cross which heals damage, a tree which produces fruit that increases strength when collected, and a hornet, an animate actor which increases the agent's damage if the agent comes in contact with it. An agent being tested in such a world must be able to find the tree and the cross, and to remember the location such that they can be found again as needed. V-World also supports worlds designed around much more complex goals, such as rescuing a princess and returning her to a throne. This sort of task can involve numerous sub-goals, necessitating an agent capable of creating and executing a plan. Worlds can also be designed in such a way that the agent does not know the effects of the various actors, which allows for the implementation and testing of learning strategies.

2.2.2 The Agents

A V-World agent has access to data concerning the contents of its immediate surroundings, as well as information about its own internal state. The agent's perceptions, or inputs, consist of twenty-four atoms that signify the contents of the twenty-four squares around it. Any given type of object or creature is always signified by the same atomic symbol. The agent also knows on each turn how much strength it has remaining, how much damage it has accumulated, what direction it moved in the last turn, and the contents of its inventory. The agent's action, or output, may be to move in one of eight directions, to sit still, or to remove an item from its inventory. These actions are also represented by a set of symbols, which are passed to the program by the agent subprogram. The agent and the other animate actors in the world make one move or perform one action on each turn. The agent code itself determines how the perceptual information available to the agent is used to decide which of the ten actions is most appropriate to take.

In addition to perceptual and state information, the agent may also be provided with further resources based on the operation of the agent's program. An agent may be designed to keep track of visited terrain by adding geographical information to its knowledge base. In the case of planning agents, the goal state, as well as information on how to order the preconditions in order to reach the goal state, may also be contained in the agent's code. Knowledge of how to properly interact with other entities in the world is generally built into the agent, unless the purpose of the agent is to learn the properties of things in the world. The agent uses the information given to it by the V-World program, combined with information stored implicitly or explicitly in the agent's decision-making process, to determine what action to return to the program as its next move. The challenge to the designer of the agent is to create a program to allow the agent to reason well enough to stay alive, learn about the environment it is in, or to accomplish whatever goals have been specified.

2.3 A SITUATED AGENT IN V-WORLD

The agent presented in the next chapter is situated in a world with a global goal that must be accomplished. The agent is designed to determine if there are preconditions that must be met before this global goal is achieved, and to order and achieve those sub-goals if they exist. The agent has no explicit knowledge, but it is composed of several levels of behavioral pathways whose configuration implicitly contains data on how to handle what it encounters. It knows the global goal of the world, as well as the characteristics of all of the atoms and actors in the world, but does not initially know the geography or what preconditions the global goal may have.

The agent's task is to rescue the princess by returning her to the throne. This may involve obtaining gold to ransom the princess from a hag that may be guarding her. Obtaining the gold may require slaying a dragon, which means finding the bane. The agent may have to slay a troll, which requires finding a sword. Some of these sub-goals can be preconditions for others. The structure of sub-goals leading to the fulfillment of the primary goal of rescuing the princess is not known to the agent at the beginning of the simulation. The agent's behavior must be structured such that it can fulfill the sub-goals in the appropriate order no matter what that structure turns out to be. Along the way, the agent must locate sources of food to maintain its strength, since the strength is decreased for every action taken by the agent.

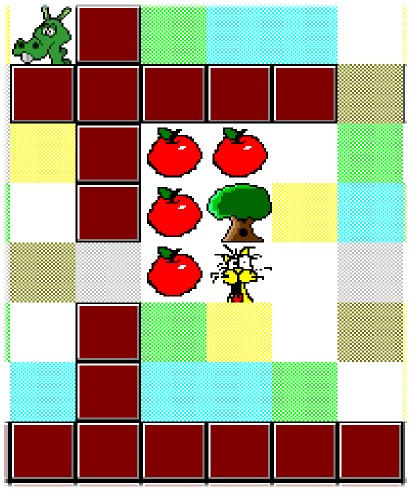
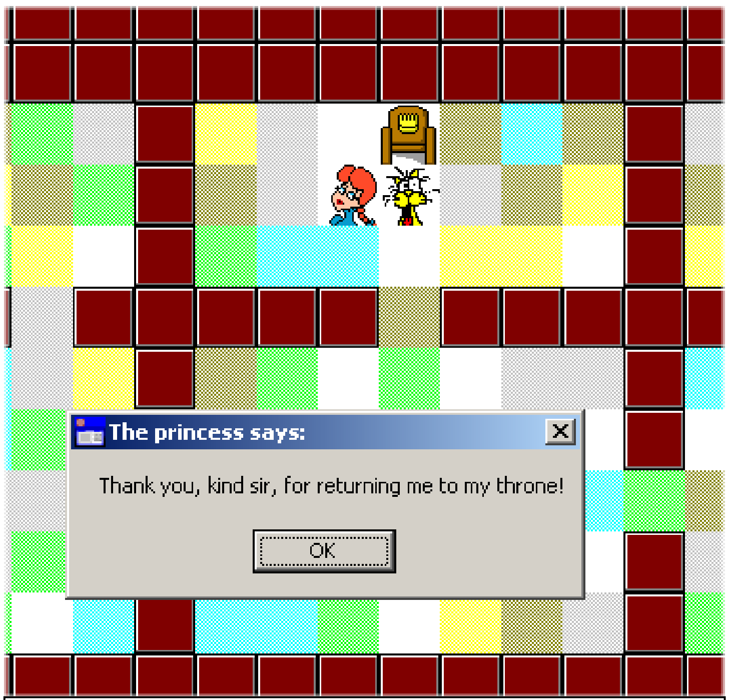


Figure 2: The agent collects apples for food

The agent must also locate a source of healing, since the presence of hostile actors in the world will lead to an increase in damage, too much of which can be fatal to the agent. In order to fulfill all of these requirements, the agent must map the terrain as it moves, in order to remember the location of important resources or dangers and to find its way back to them or away from them.

Most V-World agents take advantage of the symbolic structure of the environment to employ heuristic search as encouraged by the Physical Symbol System Hypothesis (Newell and Simon, 1976). The arrangement of the terrain of the world in discreet squares lends itself to the use of graphs or arrays in representing the geography of the world internally, as well as to pathfinding using well-established heuristic search algorithms. The categorical consistency of entities in the world, which are represented as whole symbols that always stand for the same set of properties, make advantageous the use of explicit symbolization in planning and reasoning. Several successful planning agents have been designed for the V-World planning environment that use these techniques to achieve the goals set by the domain (Bridger, Crouch, and Nute, 2000). These characteristics make the V-World environment particularly challenging for a situated agent, which by definition cannot partake of the pre-symbolization of inputs and outputs, nor of the discreet grid structure of the world itself. A situated agent also cannot make use of heuristic search, although this would be easier given the arrangement of the geography of the world, because heuristic search requires explicit symbolization of a centralized representation made use of by a centralized reasoning system. An agent based on the ideas of situated cognition is required to navigate the V-World domain using the implicit knowledge contained in layers of behavioral pathways, allowing the interaction between numerous distributed control systems to determine the correct output. Likewise, such an agent cannot store geographical knowledge in symbolic form as previous agents have done, but must add task modules whose interaction allows the agent's behavior to take geographical relationships between things in the world into account.

This agent will employ the strategies of internal situatedness, spreading activation, and behavior set modification, which will be expanded upon in the next chapter, to produce a level of effectiveness on par with the symbol system-based agents previously developed for this domain. The agent will also demonstrate the increased flexibility and fault-tolerance that arises from distributed representation distributed control. The Virtual World development environment was designed around the use of symbol system-based techniques for planning, learning, and search. The purpose of creating a situated agent for this domain is both to test a combination of navigating, planning, and decision-making strategies that have been developed for systems inspired by the Physical Grounding Hypothesis, and to show the effectiveness of the methodology itself. The V-World domain provides an ideal setting


The princess says:

Thank you, kind sir, for returning me to my throne!

Figure 3: The agent rescues the Princess

for demonstrating the potential of the situated approach to agent design because the very structure of the domain favors agents designed in accordance with the Physical Symbol System Hypothesis. It has already been established that symbol-system based agents encounter considerable difficulty in real-world domains due to the restrictions imposed by centralized control, centralized representation, and the rigid category structure inherent in explicit symbolization of inputs and outputs. By achieving the goals of the V-World domain in a more effective and lifelike manner than the symbol system-based agents for which it was designed are able to achieve, the agent is intended to demonstrate the potential that situated cognition has both to produce more intelligent artificial systems and to unravel the mysteries of human cognition.

CHAPTER 3: IMPLEMENTATION

3.1 THE PROBLEM OF NON-REACTIVE SITUATED COGNITION

A criticism frequently leveled at advocates of the Physical Grounding Hypothesis and related ideas concerns the tendency of these approaches to produce agents that are mostly reactive in nature (Maes, 1990). Researchers investigating these approaches are focusing on simple actions such as moving and navigating, rather than complex tasks such as playing chess or constructing proofs, in the hopes that these techniques can eventually be scaled up to produce more intelligent behavior. The strength of this architecture derives from creating connections from inputs to outputs that are as direct as possible, with minimal interpretation of incoming and outgoing data. As a result, researchers in other sub-fields of Artificial Intelligence have raised the question of how such a system can produce non-reactive behavior without falling back on some sort of explicit symbol-based representation of knowledge. Reactive agents can pursue only those goals within immediate perception. Anything more advanced would seem to require the same kind of data interpretation and centralized processing that advocates of the Physical Grounding Hypothesis have rejected in order to produce effective cognition over simple tasks. This concern has caused some researchers to pursue hybrid systems (Malcom and Smithers, 1990).

3.1.1 Internal Situatedness

One solution to this problem is to retain the idea of situatedness in its entirety, but to carry it further, into the inner workings of the system. The principle idea of the Physical Grounding Hypothesis is that systems should be designed around a task-based decomposition, rather than a function-based decomposition as is found in symbol system-inspired agents. This implies that every connection between sensors and actuators represents a particular behavior, saving the agent from the onus of centralized control and centralized representation. Each behavior is grounded in the agent's environment, and the agent is seen as the connection between external circumstances and the external result. However, an added level of capability can be achieved by grounding the inputs of certain behaviors in the outputs of others. Thus, for some behavioral pathways, the output from other behaviors becomes the world, or part of the world, in which they are embedded. Representation and control remain task specific, even as the behavior produced by one pathway becomes the perception of another. In this way, the idea of embedded cognition removes the boundaries of abstraction between the agent and the world, just as it removed those between input, output, processing, and storage. The agent's representation of the world extends out into the world itself, and the world with which the behaviors interact extends inward to pathways that interact with each other directly.

This meta-leveled arrangement of behaviors allows a number of behaviors the choice of activating the particular task to which they are dedicated in response to the output of a single pathway, rather than in response directly to the external world. What this achieves is to allow the agent to carry on some interpretation of the raw data, albeit in a distributed fashion that avoids the trap of centralized control and representation. A single task module can be dedicated to taking in certain information from the external world, and delivering it to the perception of a group of behaviors for whom it is specialized. Thus, complex representation and processing of data can take place without the necessity for all incoming data to be translated into an explicit symbolic language that all behaviors must share. Situatedness and distribution of representation and control are maintained, and the agent is given the capability to perform computations of greater complexity than before.

3.1.2 Spreading Activation

Another element that allows a physically grounded system to carry out behaviors requiring non-reactive goal orientation is to employ changes of state in the behavioral pathways themselves that persist over some length of time. A system can be constructed so that individual behaviors compete for the privilege of dictating the behavior of the whole agent. Such a system works in situations where a reactive strategy is most appropriate, as the behavioral pathway focused on handling a certain stimulus is more likely to activate in the presence of that stimulus and produce the action appropriate to that stimulus. The system can then be given the ability to pursue longer-term goals by introducing the idea of spreading activation (Maes, 1990). This technique, based on higher-level models of human cognitive functioning, gives behavioral pathways a parameter of duration that allows them to continue reacting to a stimulus that may not be directly perceived. Acting on some perceptions, such as the observation that one is out of milk, requires removing one's self from the situation in which the problem condition is immediately observable. Going to the store to buy more milk means that one is no longer able to directly observe the absence of milk in the refrigerator. Spreading activation allows behavioral modules to remain competitive in the action-selection process and to affect other behavioral modules for a duration appropriate to the priority of the goal represented by that behavior. This innovation enables goal-oriented actions such as creating and executing plans to take place by removing the restriction of reactivity and immediacy from situated agents. The keystones of embedded systems, namely distributed control and distributed representation, are maintained, since the agent’s decisions still arise from an architecture divided up into behaviors whose structures determine what response will arise in any given situation. Spreading activation provides situated agents with the capability to pursue long-term goals and enables them to have a computational structure sufficient to reason through meaningful sequences of actions.

3.1.3 Adding or Modifying Behavioral Pathways

A third means of endowing situated agents with the capacity for greater behavioral effectiveness involves structuring the agent so that behavioral modules can be created, destroyed or modified. While the task of enabling a system to add behaviors to its repertoire without knowing in advance what those behaviors will be is daunting, it becomes easier when the new behaviors are of a certain type and differ only with regard to a limited number of specific parameters. Adding modules to the system increases the agent's knowledge and scope of ability to interact with its environment without the need for an explicit representation language. Since behavioral pathways can interact as much or as little as is required by the designer, the process of adding new pathways needs not limit or impede the operation of existing pathways involved in other behavioral decisions.

Combined with the idea of internal situatedness and the concept of behaviors competing for activation, this technique provides greater representational flexibility to an internal environment that other pathways can interact with in order to produce effective behavior with respect to the external environment. The internal environment in which inner behaviors are situated represents, in a distributed and non-symbolic fashion, the agent's perspective on the incoming data. In the case of navigation, pathways connected directly to the agent's external perceptions can see what is around the agent, and compete to provide internal behaviors with the knowledge of whether this location is known and how it relates to other places that the agent may want to go (Mataric, 1990). Allowing the agent to modify its repertoire of behaviors provides a way to implement effective navigation in a situated agent.

3.2 THE FLOW OF INFORMATION THROUGH THE AGENT

The agent described in this paper is embedded in a world consisting of symbols. Perceptual data and information about the state of the agent is given to the agent in the form of alphabetical or numerical symbols, and the program that constructs the world looks to the agent program for alphabetical symbols representing the agent's chosen behavior. In order to incorporate the advantages of distributed control and distributed representation, this agent uses a distributed perception system and a distributed acting system. This allows the agent to provide behavioral pathways with data about the world that is unsymbolized, and can therefore be interpreted by the interaction of those pathways, rather than by formal operations over symbolic expressions.

The external environment in which the agent is embedded consists of the contents of the world that the agent inhabits, the agent's state of health, the list of objects in its inventory, and the interaction of goal modules that are pre-programmed according to the task that the agent was designed to achieve. All of these factors make up the world in which the agent is situated. The goal modules can be considered part of the agent, but they are also sources of information about the world. Because of this, they can also be considered a part of the environment, particularly by the inner behavioral pathways of the agent.

The agent's decision as to which of eight possible actions to take is made by a set of ten behaviors that compete with each other for activation. Each of the ten is fed activation by other behaviors on the basis of where the agent is, what is around the agent, and where the agent needs to go to assure its survival or to fulfill the goals of the domain. All of the factors relevant to the decision are evaluated at once, as each exerts a pull on the agent in one or more of the eight possible directions. This continues until the activation level of one of the eight output behaviors passes a predetermined threshold, at which point it activates. In this way, the agent's decision is a vector sum of all of the forces exerted by active behavioral pathways during a turn. The task of these eight behaviors is to send a symbol representing one of eight directions to the V-World program, which moves the agent and updates the world accordingly.

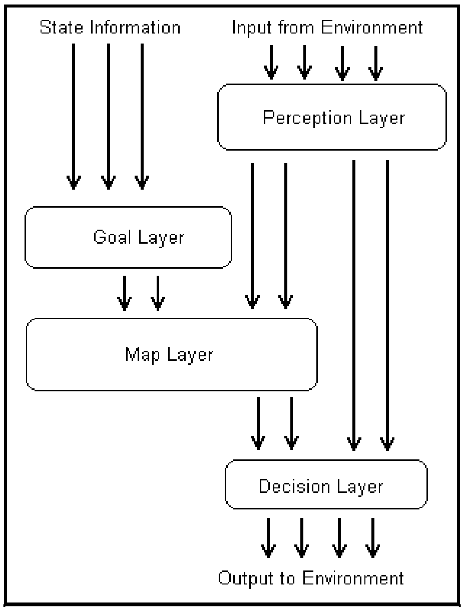


Figure 4: The flow of information through the agent

The agent can be viewed as a collection of behavioral pathways, that interact in order to determine what the agent's action will be. The next several sections explain how the agent uses awareness of its internal and external environment to make this decision. The idea of layers is not a part of the design philosophy, but is a convenient way to explain the functioning of the agent's various behaviors as they interact in the decision-making process.

3.2.1 Perception/Input Layer

The task of the behaviors in the perception layer is to take in data from the environment and distribute it appropriately to other behaviors. The V-World domains for which this agent was designed are made up of combinations of twenty object types. This list includes seven terrain objects such as walls and empty space, hostile animate actors such as the troll, the princess that must be rescued, and objects necessary to the agent's goals, such as the tree and the throne. The agent views the objects that make up its environment twenty-four at a time, essentially viewing a five-square-by-five-square section of the world with itself at the center. The perception layer is a bundle of 480 individual behaviors, each of which represents the presence of a specific object at one of the twenty-four possible positions in the agent's perceptual field. For each of the twenty-four squares that the agent can see, there are twenty nodes, each firing if the object type it represents is occupying that square. One such behavior, for example, might become

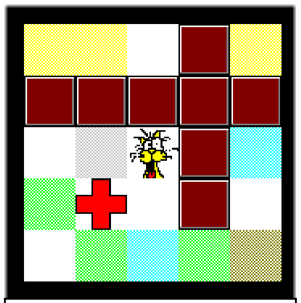


Figure 5: The agent's 24-square visual field

active if there is a wall in the upper left corner of the agent's field of vision, while another might activate when there is a troll directly to the right.

On each turn, the agent observes the environment, and twenty-four perceptual pathways become active. The process of observation by the agent is just the process of one percept node becoming active for each of the twenty-four observable squares, to tell the rest of the agent's pathways what object is in that position relative to the agent. The perceptual pathways are essentially binary, remaining dormant if the object they represent is not present in their space, and activating other processes down the line if it is.

One of the roles of the perception layer is the handling of reflex actions. The agent's distributed control architecture precludes central controls that clamp down on the reasoning process when a quick decision is required. Since each decision by the agent is a vector sum of all of the agent's concerns at that point, it is possible for the agent to react incorrectly to an immediate threat or opportunity due to indirect influences pulling it in other directions. The agent's reflex behaviors exist to counteract this tendency. Perceptual pathways representing objects that present the agent with a threat, such as trolls and dragons, or opportunities, such as the tree or the gold, are connected directly to the eight action-generating behaviors in such a way as to strongly influence the agent's decision and to drown out influence from other concerns.

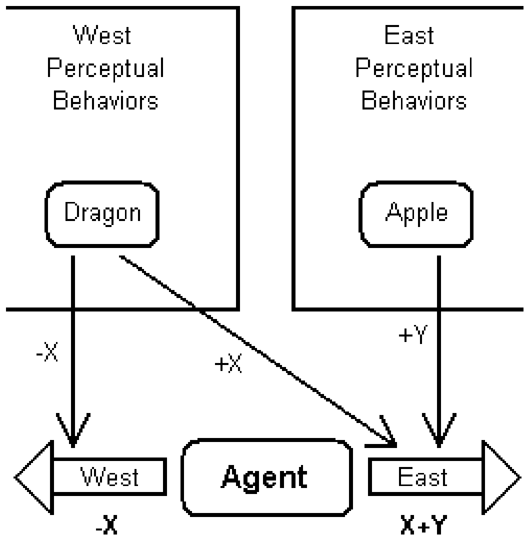


Figure 6: Perceptual behaviors activate the appropriate decision behaviors, guiding the agent away from the dragon and toward the apple

If the agent detects a troll immediately to the left, the perceptual pathway that responds to the troll in that position sends a large amount of activation to the output behaviors representing motion in the opposite direction. If the agent needs the gold to ransom the princess and the gold is directly above it, the percept node representing gold in that position directly activates the output behavior corresponding to motion in the direction of the gold. This strategy allows immediate concerns to raise a preferred direction above the decision threshold before less critical or direct influences have a chance to dilute or outcompete this preferred direction. Thus, the agent can focus on immediate concerns when necessary, without the need for a central executive module to adjudicate between behaviors.

The perception layer is also responsible for making information about the agent's surroundings available to the behaviors that determine which goals to pursue, and those that determine where the agent is. The percept nodes that make up the perception layer are linked to nodes in the map layer and the goal layer. The goal layer uses perceptual information to determine if anything in the agent's immediate surroundings is needed for survival or fulfillment of one of the agent's current goals. The map layer uses perceptual information to determine whether the agent's location matches a place the agent has been before. Perceptual information is used to update the behavioral representation of a given location if there have been changes since the agent was last there, or to create new navigation behaviors if the agent is in a new location. The agent program itself provides links between the percept behaviors and the appropriate behaviors in the goal layer and the map layer.

3.2.2 Goal Layer

The function of behaviors in the goal layer is to keep track of concerns beyond the agent's immediate perception. The goal layer provides the agent with another source of information about the state of the world by combining the agent's observations over time with knowledge about the goals of the domain. The output of perceptual behaviors help to determine which goals are active at any given point, and dictate the strength of that activation in comparison to other influences on the decision-making process. The agent's strength, damage, and inventory also provide information to the goal behaviors. Each behavioral pathway in this layer represents the desire to perform a certain task, and each task competes with the others for activation which is used to influence the spreading activation process in the map layer, or the competition between the output behaviors that determine the agent's actions. This goal resolution architecture is inspired by Maes' work in the use of spreading activation for goal resolution in situated agents (Maes, 1990).

Factors relevant to the agent's survival in the world take the form of goals, and are handled by the goal layer along with the rest of the agent's long-term priorities. Some survival concerns, such as avoiding nearby enemies and partaking of immediately available resources, are governed by the reactive behaviors in the perception layer. However, tasks such as moving to a distant tree to obtain fruit, traveling to a cross to be healed, and tracking down an enemy to destroy it in order to provide a safe path for the princess cannot rely on immediate perception to keep the agent focused. Survival tasks that require more than reactive thinking on the part of the agent are structured as behaviors in the goal layer, in the same way as the tasks that the agent must perform in order to be successful in the domain. This arrangement allows the agent to prioritize maintenance activities on the same scale as goal-related activities. It also provides a means for governing reactive behavior. The agent should use the tree to obtain fruit if the tree is nearby, but there must be a point at which the agent decides that it has eaten enough. Otherwise, it is possible for the agent to spend all its time eating, to the detriment of its goals. The amount of activation accumulated by survival-related goals determines how reactive the agent is to immediate threats or opportunities. While life-or-death situations are best handled reflexively by the perception layer, the goal layer allows for in-between states where reactive thinking may not be necessary, but is beneficial up to a point. If the agent is on its way to rescue the princess, and it passes near a tree, it makes sense to obtain some food, but not to abandon the previous goal. This arrangement of survival concerns allows for this flexibility of action by the agent.

The goal layer is also responsible for determining what sub-goals must be fulfilled in order to satisfy the primary goal of rescuing the princess, and deciding in what order to fulfill these sub-goals. Rescuing the princess may involve paying gold to the hag, and obtaining gold may require slaying a troll or a dragon. Defeating these hostile actors requires the sword and the bane, respectively. This presents the agent with a range of possible scenarios, in which an unknown series of sub-goals must be performed in the proper order. To fulfill the requirements of the domain, the agent must be able to discover what the goal structure of the particular situation is, and must be flexible enough to rescue the princess no matter what the combination of sub-goals turns out to be. The goal layer provides the agent with this flexibility by implementing each goal as a behavior that interacts with other goal behaviors. These goal behaviors become active, and activate or inhibit other goals, based on state information or observations by the agent.

For example, when a dragon is nearby, the perception behaviors responsible for detecting dragons become active, and activate the goal behavior dedicated to avoiding the dragon and finding the bane. When not being activated by perception behaviors, the dragon-handling goal behaviors activate the dragon-detecting perceptual behaviors to see if these behaviors have formed links to location behaviors. If they have, then the agent has encountered a dragon, and the dragon-handling behaviors activate the goal behavior dedicated to finding the bane. If the bane is currently in the agent's inventory, the bane-finding goal behavior inhibits dragon avoidance, and activates dragon hunting. Goal behaviors relating to the dragon and other such threats can also be influenced by input from the agent's observation of its own level of damage.

This arrangement similarly governs the agent's central goal of returning the princess to the throne. The agent is constantly motivated by the princess-rescuing goal behavior to go to the throne once it encounters the princess. However, if the agent detects or has in past turns detected a hag, it activates the links between hag-detectors and princess-detectors to see if the position-representing behaviors that the two link to are adjacent. If this is so, then the agent is influenced to seek the gold. If the agent encounters a dragon or a troll on the way to the gold, it is influenced by other goal behaviors to seek out and make use of the

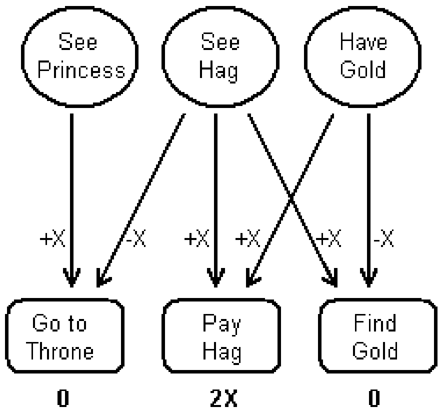


Figure 7: When the agent sees the princess and the hag and has the gold, it is motivated to pay the hag

appropriate tools for defeating them. If the agent already possesses the gold upon encountering the hag with the princess, the agent is influenced by the hag-paying goal pathway to pay the hag and ransom the princess.

In this way, the interactions of the agent's goal behaviors determine what influence goal-related concerns will have on the agent's decisions. Contradictory goals inhibit each other, goals that depend on sub-tasks activate those sub-tasks, and survival-related goals activate or inhibit each other depending on the magnitude of the threat to the agent's well-being. The goal behaviors can either be activated by the output of other behaviors, or can generate their own activation which allows them to activate other behaviors and influence the output of the agent accordingly.

Because the control architecture of the agent is distributed rather than centralized, several goals may be active at once. In order to influence the decision process, a goal behavior must achieve an output that equals or surpasses a certain threshold. Those that do not pass the threshold do not act on the mapping layer, where decision-making takes place. Any number of goals that do pass the threshold, however, may be simultaneously active. All active goal behaviors that pass the threshold send activation through links to the decision layer, where they affect the interactions of navigating behaviors in order to influence the agent's decision as to where to move during each turn. The goal-resolution is partially carried out during the interaction of the goal behaviors, prior to the imposition of the threshold. Those that clear the threshold compete via spreading activation with influences from the perception layer to guide the agent's course.

The agent's goal behaviors have links to behaviors in the mapping layer that allow them to affect the decision-making process. Each goal that passes the threshold injects its activation into the mapping layer, which is organized geographically. The representation of the appropriate change in the agent's behavior to bring about each goal is contained in the links between the goal layer and the mapping layer. These links ensure that activation from the goals is routed to the appropriate behavioral pathway in the agent's representation of the terrain, to guide the agent's course toward objects or locations that will enable the fulfillment of that goal. When several active goals pass the threshold, each injects its activation into the appropriate part of the mapping layer, where the goals compete by making certain areas of the map more active and influencing the agent to move to these areas. The activation from goals that pass the threshold competes and interacts with activation from other behaviors to influence the agent to move in the direction that simultaneously satisfies as many goals and concerns as possible.

3.2.3 Mapping/Navigation Layer

The mapping layer of the agent handles the dual functions of representing the terrain the agent has visited, and serving as the arbitration medium for influences from the perceptual and goal behaviors. This layer is made up of behaviors representing positions, or discrete geographical spaces that the agent can occupy, and moves, which carry the agent from one geographical space to another. This strategy is inspired by Mataric's work in distributed landmark-based navigation (Mataric, 1990).

The position behaviors represent all of the spaces in the world that the agent has occupied. Each position behavior is linked to up to twenty-four perceptual behaviors. This is the set of perceptual behaviors that were active when the agent first occupied this space. Each time the agent moves, one perceptual behavior becomes active for each of the twenty-four squares that the agent can see. The perceptual behavior that becomes active for each square represents the presence in that square of one of the twenty object types that make up the world. Each active perceptual behavior then sends activation down links to position behaviors that it was previously associated with. The position behavior that receives the most activation above a threshold of twenty is assumed to represent the agent's current location, and this position behavior becomes active. The function of the active position behavior is to determine which eight move behaviors represent moves that the agent can actually make from where it currently is. The position behaviors associate a set of objects with a location in the agent's distributed internal map. They also allow other behaviors, such as goal pathways, to determine where on the map a given object type is located.

The move behaviors represent moves that take the agent from one position to another. Each move behavior has its origin in a position behavior and its destination in a different position behavior, a stationary object type, or the frontier behavior. Those that terminate in stationary objects, such as trees or crosses, represent moves that do not change the position of the agent. Those that terminate in the frontier behavior guide the map updating process, and serve as targets for activation from the exploration goal pathway. The move behaviors receive activation from perceptual behaviors, goal behaviors, and from other move behaviors. Each move behavior outputs activation to the eight move behaviors whose destination position is the same as the activating move behavior's position of origin in most cases, or outputs to one of the eight decision behaviors if the activating move node is one of the eight surrounding the position behavior representing the agent's current location.

Navigation presents a particular challenge to designers of situated agents, because we as humans are accustomed to thinking about geographical knowledge in the language of symbols. Conscious cognition in humans frequently takes a form that is better suited to communication than to action, and thinking about locations and how they relate to each other is no exception. However, geographical knowledge can also be represented behaviorally, demonstrated by the fact that most of the navigating we as humans do is done without conscious thought or reference to explicit directions or images. Moving from one's kitchen to the living room is done with the same sort of unthinking familiarity with which one plays piano or rides a bicycle, suggesting that the knowledge is stored in behavioral pathways that operate with no need for explicit symbolization. This kind of behavioral navigation can be implemented in a situated agent by creating new behaviors to represent areas of the terrain. Combined with the strategies of internal embeddedness and behavioral competition as described above, this technique results in an agent possessing a number of navigational behaviors that compete among themselves to represent the agent's current location.

The process of deciding where to move begins with input into the mapping layer from the goal and perception layers. The input layer injects activation into the eight active move behaviors in order to influence the agent in response to immediate threats or immediate opportunities. This can mean, among other possibilities, activating the pathway representing motion to the north to avoid a troll immediately to the south, or activating the pathway to move southwest to collect fruit directly to the southwest. The goal pathways can activate move nodes that terminate in a necessary resource, such as a tree, sword, or cross, wherever the resource is located, or move nodes that terminate in a position behavior that is linked to perceptual behaviors representing something important, such as the princess or the throne. The non-active move behaviors, which are not adjacent to the agent's current position and therefore cannot directly activate the decision behaviors, spread activation backwards to move behaviors whose destination matches the origin of the activating move behavior. The level of activation of a move behavior represents a desire to make that particular move. When the agent is hungry, the move behaviors terminating in trees become active, demonstrating the agent's intention to push against a tree to produce fruit. This move behavior, in turn activates move behaviors that would result in the agent being in a position that would make that move possible, representing the agent's desire to move to a point from which it can make use of the move to the tree. In this way, activation spreads backward until one or more of the active move nodes, the eight that surround the agent and are candidates for activating a decision behavior, receives activation from it. This process occurs in cycles, with each move behavior that has an activation greater than zero passing its activation backwards to its neighbors once during each cycle. The goal and perceptual behaviors also re-inject their activation at the beginning of each cycle. After some number of cycles, the activation level of one of the eight active move behaviors will pass a threshold, at which point it activates a decision behavior which outputs the direction that it represents to the world program itself.

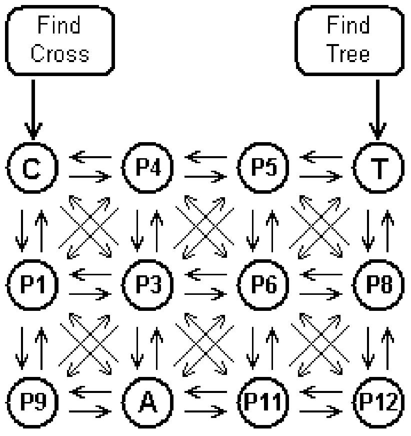


Figure 8: Tree and cross-finding goal behaviors (T & C) activate position behaviors, which spread activation toward the agent (A) through move behaviors

The effect of this strategy is to produce concentric waves of activation spreading out from locations on the internal map to which the agent should travel. These waves eventually reach the agent's perceived location on its internal map, at which point the activation combines with reflexive motivation delivered directly to the active move nodes from the perception layer. The fact that multiple goals and multiple reflexive perceptual impulses can be simultaneously active allows each decision that the agent makes to be a vector sum representing the combined influence of all of the agent's current concerns. This allows the agent to be both efficient and opportunistic. If the requirements for fulfilling two goals lie to the west, and the object necessary for one goal lies to the east, the activation from the west will overlap and combine, encouraging westward motion. If the agent is somewhat hungry and passes near a tree while en route to another location, it will detour slightly to fill up on fruit before continuing, rather than traveling to its destination and then traveling all the way back to the tree. If multiple routes exist to a given objective, the agent will be encouraged to take the one that passes near unexplored territory, or a food source, or any other objective that can be completed on the path to another. The capability to evaluate multiple goals simultaneously and to make decisions that fulfill multiple goals simultaneously is one of the principle advantages of a distributed decision-making architecture.

The agent adds behaviors to the map layer in situations where no position behavior passes the threshold activation necessary for it to be considered the agent's current location. This occurs when the agent moves into a space that it has not previously occupied, or when it moves into a space that has been previously occupied but has changed significantly. When no position behavior is able to become active, a new position behavior is generated from a template and is linked to the twenty-four percept nodes currently active. This process associates the new location behavior with the objects observed around it, so that the agent will be able to identify the location if it is encountered again. Move behaviors are then generated to and from any adjacent locations that are already represented as position behaviors. The agent determines whether adjacent locations are on the map by re-activating the perceptual pathways that the current location has in common with the adjacent location. If the adjacent location passes a threshold, lower than that used to determine current location, then move nodes are created to and from that location to reflect the agent's ability to traverse from one to the other. If the adjacent location is not identified, a move behavior is generated from the current location that has the frontier behavior as its destination. When the exploration goal pathway is active, all move nodes with frontier as a destination receive activation from it, and spread that activation through the network of move behaviors to guide the agent toward unexplored territory. The exploration goal pathway is active by default, and is inhibited when a goal behavior related to the other goals of the agent becomes active. This means that when the agent does not know the location of resources necessary to fulfill the current goal, the exploration goal pathway becomes active and drives the agent to expand its representation of the terrain.

Some of the objects in the agent's environment are animate actors that move around, which requires that they be handled differently by the agent's map updating process. Objects such as the princess, the hag, the troll, and the dragon activate perceptual pathways dedicated to them, which link to the map updating process in such a way that the squares occupied by these animate actors are mapped as empty traversable space, one of the other terrain object types. This means that the agent will be unable to perfectly represent the terrain observable from certain positions. However, the consensus strategy used to determine current location allows for a certain margin of error. The agent is able to determine its location even if not all of the same perceptual pathways are active as there were the last time the agent visited that space. When this is the case, the agent updates the links to a position node to reflect the activation of perceptual pathways currently in use, to keep the map up to date and as correct as possible. The ability to make decisions with partial or less-than-perfectly-reliable data is a key benefit of distributed navigation strategies.

3.2.4 Decision/Output Layer

The function of the decision layer is to interact with the V-World program by outputting a symbol representing the agent's chosen direction. Each of the eight decision behaviors is linked to one of the eight active move behaviors adjacent to the agent's representation of its current location. When one or more of these active move behaviors passes a preset threshold, the one with the highest activation enables the decision pathway to which it is linked. This pathway then returns a directional symbol to the V-World program. The purpose of the decision layer is to interact with the symbol-based architecture, allowing the rest of the agent to operate as much as possible without recourse to symbol manipulation.

3.3 CONCLUSION

The implementation of the concept of internal situatedness provides this agent with a means for filtering and interpreting external data without the need for logical operations on symbol sets or monolithic representation. The perceptual pathways are dedicated to specific aspects of the environment, which allows them to organize information about external conditions for delivery to other pathways that are designed to react only to certain circumstances. The perceptual behaviors make up the environment in which behaviors in the map layer and the goal layer are situated. The goal layer itself provides part of the environment external to the map layer, which in turn acts as the medium in which the decision layer is embedded. The concept of situatedness or embeddedness is extended into the agent architecture itself, allowing for greater specialization of behavioral pathways, and a greater variety of information handling possibilities for the agent as a whole. Behaviors do not always have to originate and terminate in the environment external to the agent. Internal situatedness allows the essence of situated or embedded cognition to be preserved, along with the benefits thereof, while granting a greater range of behavioral capabilities to the agent.

The use of spreading activation allows the agent to continually provide inward behavioral pathways with goal-related activation, even when goal related stimuli are unavailable to externally embedded pathways. This technique is a key element in overcoming the barrier of reactivity that is frequently thought to be a limitation on embedded agents. Spreading activation provides the agent with a form of representation that persists over time, and can be used to maintain goal-directed behavior in the absence of immediately observable triggers in the environment. This concept also provides situated agents with a substitute for heuristic search techniques, allowing agents to engage in pathfinding computations without the need for explicit representation or symbolic techniques. The pathfinding technique employed in this agent finds not only the shortest valid path to any given goal, but the path that encompasses as many goals as possible simultaneously. Spreading activation both provides situated agents with pathfinding techniques equal to symbol system-based agents, and offers the added bonus of effective resolution among multiple goals.

The agent is able to overcome the limitation of a static body of behavioral knowledge through modification of the set of behavioral pathways. The perceptual pathways form new links to behaviors representing geographical position in order to associate external landmarks with internal information about the relative positions of objects in the world. The agent also creates new position and move behaviors to represent paths that can be taken from place to place, and to create a medium for decision-making and pathfinding via spreading activation. The capability of adding to the existing behavior set gives the agent the capability to reason about areas of its environment that are not immediately perceivable. Storing geographical knowledge in the form of behaviors that interact with each other and with other elements of the agent gives the agent the ability to use this knowledge without symbolic operations such as heuristic search, and to find multiple paths or paths to multiple objectives in parallel.

The strategies implemented in this agent are intended to allow it to display goal-directed behavior, to make effective decisions involving multiple concerns, and to successfully navigate a varied and dynamic environment without recourse to explicit symbolic representation or centralized control. This experiment is meant to demonstrate the capability of situated agents to rise above the limitation of simple reactivity, by substituting an ontology of behaviors for an ontology of symbols. The next chapter presents the results of the implementation of this architecture, and provides some discussion of the implications of these results.

CHAPTER 4: RESULTS AND CONCLUSIONS

This chapter concerns the results of the implementation and execution of the agent. Discussions follow concerning design choices made during implementation, details of the agent's performance, and problems encountered during execution. The conclusion is a summary of the agent's success both in the achievement of the goals of the domain, and in the demonstration of the effectiveness of situated cognition as a model for agent design.

4.1 DESIGN CHOICES

This agent is intended as a demonstration of the effectiveness of agent architectures based upon the principles of situated cognition. When an agent inspired by these principles is physically manifested, such as in a mobile robot, it is possible to build an architecture that is actually parallel and completely embedded in its environment. This project, however, concerns a simulated agent operating in a simulated environment made up of discrete, explicit symbols. This being the case, not every element of situated cognition is implemented explicitly. The next few sections detail departures from the ideal of embedded design that became necessary due to environmental or computing constraints. The agent itself holds true to the intent of situated cognition in essence, but the following deviations should be noted.

4.1.1 Implicit Parallelism

The serial nature of the physical computers available to run this agent in V-World proved to be a significant design constraint. A situated agent is in essence a parallel computational architecture, and implementation of parallel architectures on computers with serial processors requires that the calculations be performed one at a time to reproduce the effect of parallel computation. Ideally, the structure of the agent code would be identical to the theoretical structure suggested by task-based decomposition. Each behavior would be represented as a module of the program, and each could be processed in turn to produce the appropriate output. However, the interconnectedness of the behavioral modules themselves means that they cannot be activated one at a time, or they will have no opportunity to interact with each other during computation. It was therefore necessary to represent some of the behavior modules implicitly, interleaving the lists of commands that made up the computational structure of the modules themselves in order to ensure interactivity.

Each goal behavior, for example, is split into an input segment and an output segment. The input segments of all of the goal behaviors are processed as a group, so that each can determine based on perceptual and state information whether or not it will become active. Each of the input segments that becomes active then calculates the excitatory or inhibitory effect that it will have on other goal nodes, and to what extent it will activate its own output segment. In the second phase of goal activation, the excitatory influences, inhibitory influences, and activations are applied to the appropriate goal output segments. The output segments are then checked to determine which, if any, pass the goal threshold. Those output segments that pass the threshold then exert their influence upon the map layer to guide the decision making process.

4.1.2 Low-Level Centralized Control

In this implementation, the behavioral pathways are represented by data structures that are acted upon by the agent program to simulate a completely distributed control architecture. While it would have been more structurally accurate to represent the pathways as executable modules, it would have been difficult to achieve the necessary parallelism and interaction between modules using this method. The essence of situated agency is preserved by the fact that the flow of information is organized in terms of behavioral pathways, rather than in terms of component modules dictated by a functional decomposition. The patterns of the agent's behavior demonstrate that the benefits of the situated design strategy do arise despite the separation between the knowledge possessed by the agent and the control structures that make use of it.

4.1.3 Symbolic Input to Goal Behaviors

In the description of the flow of information through the agent, the goal behaviors are described as drawing upon the input of the behaviors in the perception layer. In the implementation of the agent, however, the goal behaviors directly query the symbolic input from the environment. Since the environment itself is made up of discrete symbols inhabiting discrete geographical spaces, this change makes no difference in how the agent operates. This alteration to the original plan was made to save processing time, which is of significant concern given the computational burden of running simulated parallel decision-making structures on a serial device. The goal nodes themselves employ distributed interaction to adjudicate among themselves, so the spirit of the project is implicitly preserved, if departed from explicitly.

4.2 PERFORMANCE OF THE AGENT

Agent architectures based on the principles of situated cognition offer several benefits that cannot be practically implemented in symbol system-based agents that rely on centralized control and centralized symbolic representation. Among these benefits are decision-making oriented toward several goals simultaneously, life-like patterns of behavior, graceful failure, and greater flexibility in the face of incomplete or incorrect information about the environment. Traits such as these are a primary motivation for research in situated cognition and embedded agency. Designing systems with these abilities will both improve the usefulness of technology based on autonomous agent architectures and our understanding of the workings of intelligence in natural agents. The following sections describe the performance of the agent presented here from the perspective of these four traits, with the goal of showing that the agent not only completes the task for which it was designed, but also displays characteristics that demonstrate the potential of situated cognition.

4.2.1 Multiple Goal Orientation

The agent's behavior demonstrates the ability to make decisions that fulfill multiple goals at the same time. In instances where the agent is confronted with a threat such as a troll or dragon, the agent is compelled to move to one of the three spaces in the opposite direction. The agent's choice among the three is made according to other goal influences. The agent moves away from the troll in the direction of a tree if it is hungry, or in the direction of the sword that will allow it to destroy the troll. This tendency is also displayed in the agent's route to distant objectives. When the agent catches sight of the princess and the hag, the gold-finding behavior becomes active. Since the gold is usually located at the other end of the world from the princess and the hag, the agent must plot a course across the world. If the agent is damaged, its travel path tends to take it to the cross that is located slightly off the path. If the agent is hungry at all, the path between the princess and the gold will be bent toward the tree, where it can replenish its strength. Behaviors such as these indicate that the agent is capable of working to achieve multiple goals at once. This is made possible by the fact that all of the active goal behaviors compete to influence the vector sum that determines the agent's next action.

4.2.2 Life-like Behavior Patterns

The behavior of the agent also displays greater similarity to that of natural agents than previous V-World agents have shown. When the agent is located between two unexplored areas, or is equally drawn to two resources that are currently needed, it sometimes oscillates back and forth for a few turns before making its way to one of the alternatives. While this does not contribute to the agent's effectiveness at performing the tasks for which it was designed, it does suggest that the distributed decision-making architecture may hold promise as a tool for understanding natural cognition. When the agent's priorities change due to some new internal or external circumstance, the agent does not immediately discard previous priorities, as symbol system-based V-World agents tend to do. Instead, the new priority adds its influence to the decision-making process, and the agent's movement is biased in that direction. If the new priority changes the agent's direction or focus, other priorities will still remain active and attempt to guide the agent to resources that satisfy their demands en route to the new destination. Previous V-World agents tend to pursue one goal at a time, completely ignoring previous concerns if a new goal becomes relevant. The greater adherence to natural patterns arising from this architecture may mark it as useful in the development and testing of theories concerning cognition and decision making in humans and animals, leading to an increase in the understanding of intelligence as a whole. The ability of this design to evaluate and pursue multiple goals simultaneously may also have applications in the field of intelligent control systems for dynamic environments.

4.2.3 Graceful Failure

When the agent experiences a malfunction in one of its components, or finds itself in circumstances not foreseen by the designer, it does not fail completely as other agents are prone to do. Instead, the parallel design of its decision-making architecture allows it to determine an action no matter what the conditions. The action may not be the best choice, but the agent will continue to operate. This characteristic, known as graceful failure, is a much sought-after virtue of embedded systems, particularly those designed for dynamic or noisy environments. Typical systems rely on the ability to make black-and-white distinctions when evaluating the environment. They also tend to handle noise poorly, and depend upon every sub-component in the organization of their internal hierarchies functioning perfectly (Mataric, 1997). The ability to handle failure gracefully means that the agent can recover from significant external change or internal malfunction to the extent that it is able to continue functioning to the best of its ability. The binary distinction between working and not working typical to most artificial systems is replaced with functionality that is roughly inversely proportionate to the extent of the change or mishap. If one part of the agent ceases to work, other parts carry on. This agent displayed this capability during the debugging process, once the decision-making infrastructure was in place. When one or more behavioral pathways were malfunctioning or inactive, the agent continued to move around in the world, using the set of behaviors that were unaffected by the current bug. When an incorrectly functioning behavior sent the wrong type of signal to the decision-making process, the outputs of behaviors dedicated to other goals were unaffected. The only problems capable of completely stopping the agent were those that occurred in the control processes that governed the behaviors themselves. Since the purpose of these processes is to simulate a physically manifested parallel control structure, they would not be present in a truly embedded system. The fact that these processes are the sites of the agent's only catastrophic failures demonstrates the strength of the parallel architecture that they support.

4.2.4 Flexibility

The grounding of the agent's representations in the environment in which it operates provides the agent with a greater degree of behavior flexibility when dealing with unknown circumstances, incorrect data, or changes in the environment. The implicit distributed representation scheme frees the agent from the need to rely on certainty or predictability. The agent's environment contains animate actors that move around, as well as collectible objects such as the sword, the bane, and fruit from the tree that disappear when the agent picks them up. This means that the agent's representation of locations by the network of mapping behaviors is not always completely accurate. Enough change to a given location means that the agent may be unable to recognize it. When this happens, the agent is able to create new behaviors to represent the modified location, and attach them appropriately to the existing network so that the collection of mapping behaviors continues to function as a decision-making component.

The agent displays similar flexibility in the handling of hostile animate actors. Links between perception behaviors dedicated to sensing trolls and position nodes activate behaviors dedicated to finding the sword, then destroying the troll so that it will not present a threat to the princess. To find the troll once the sword is in its possession, the agent activates the perception-to-location links to activate the location behavior where the troll was last seen. If the troll is not there, the agent is able to wait until the troll appears again to pursue and destroy it. In the version of the planning domain used to test the agent, the troll guards the bane, which is used to repel the dragon. This arrangement means that finding the sword with which to destroy the troll can become a sub-goal of obtaining the bane with which to defeat the dragon. However, in one trial, the agent saw the opportunity to slip past the troll and get the bane. Having seen the troll, the agent later obtained the sword and defeated it, but it was able to rearrange its goal-oriented behavior to reflect the achievement of a goal despite not fulfilling what had been a sub-goal.

The agent further displayed its capability for flexible behavior in a later run, when it caught sight of the hag guarding the princess on the other side of a wall that the agent couldn't cross. Once the agent saw the hag, its gold-finding behavior became active, leading it to the gold. After obtaining the gold, the agent returned to the spot from which the hag was visible, despite the fact that there was no way to get to the hag from that spot. However, once this became apparent, the agent was able to shift its priorities to finding the princess, who was not visible from that location. The agent took another route to the princess, encountered the hag, and paid the ransom. The agent was then able to lead the princess to the throne. The agent's flexibility in working toward solutions to its goals allowed it to circumvent a problem area not anticipated even by the designer.

4.3 PERFORMANCE PROBLEMS

The agent completed the goal of rescuing the princess successfully in most trials. The agent was able to adjudicate between goals, to execute necessary sub-goals in the proper order, and was successful at navigating the domain and maintaining a usable idea of its location and the locations of other objects and entities in the environment. A few problems arose, however, that may be the focus of further work on the subject.

4.3.1 Slow Execution

The most prominent difficulty encountered during testing was the slowness of the agent program itself. Even on a computer equipped with an 850-Megaherz processor, the completion of successful trials took over an hour, where typical V-World agents are able to complete the domain requirements in roughly ten minutes. The agent's computational architecture is parallel in structure, but this is achieved through a simulation that is executed on a serial physical computer, as most desktop computers are. While a physically manifested parallel architecture would avoid much of this problem, parallel computing devices are far behind serial systems in terms of sophistication and development. The time taken by the agent for each move tends to increase consistently with the size of the agent's internal map. Each new location that the agent remembers means another nine behaviors added to the decision making process, and every behavior is necessary for the agent to evaluate all currently relevant factors. Decisions went from taking less than a second at the beginning of each successful trial to taking as much as thirty seconds by the end of the trial. This implies that the focus of the problem is in the representation of each location square in the environment as a set of programs that interact with the rest of the agent.

Reducing the amount of code used to represent the behavioral pathways did speed the process up somewhat. Changing the numerical representation of signals within the agent from floating-point to integer also sped up the decision-making process. Further speed was achieved by not running pathways in the decision-making layer that had not exceeded an activation level of zero, which had no effect on the agent's effectiveness since modules with zero activation were not a part of the evaluation process. Much of the time expended does seem to be due to the problem of running a parallel decision-making program on a serial machine. The slowness of the agent program remains a practical difficulty, but does not reflect on the usefulness or validity of the ideas manifested in the design of the agent.

4.3.2 Oscillating Between Goals

The oscillation mentioned in the discussion of the agent's performance above is somewhat of an issue concerning the effectiveness of the agent. While the agent does eventually succeed in the achievement of the goals of the domain, it occasionally spends ten or twelve turns moving back and forth between two goals of roughly equivalent influence. These cycles end when the level of influence for one of the goals changes, or when another goal passes the goal threshold and begins exerting its influence on the decision-making process, or when some change in the environment presents the agent with a new threat or opportunity. For example, when the agent is exploring, it may vacillate between two unexplored areas of equal size and distance from the agent. If it does this for long enough, its hunger increases to the point at which its decision-making process is affected, and it moves toward the tree, breaking out of the dilemma. The problem is partially caused by the manner in which activation spreads through the decision-making network of behaviors. On each activated move behavior spreads a small percentage of its activation to the eight move behaviors terminating in its point of origin, to represent encouraging the agent to move to the point where that move behavior becomes viable. This means that a given move behavior between the agent and a relevant goal location receives activation from the nodes between it and the goal, and from nodes to either side of that path that have also become active. Thus, if an agent moves toward a goal, the activation toward that same goal will seem to build more slowly, and may be overcome by the activation from a goal in the opposite direction. This phenomenon contributes to the oscillation. The problem can also be considered the result of a life-like decision-making architecture, in the manner of the proverbial donkey starving between two piles of hay.

This problem may be solvable by changing the manner in which the move behaviors activate each other so that the total influence on the agent from any goal is more a function of distance than of time. It might also be possible to add a behavior or strategy that causes and maintains fixation on one goal in cases where two competing goals are judged to be of nearly equivalent importance. Regardless, this problem occurs infrequently, in a very specific set of circumstances, and has not been shown to detrimentally affect the agent's performance to a great degree. It is, however, an area of possible improvement for future attempts along these lines.

4.3.3 Getting Cornered by Hostile Actors

The most significant performance issue that has arisen thus far is the agent's tendency to get into situations involving hostile animate actors from which it cannot escape. The agent's motivation to explore often takes it into confined spaces guarded by the dragon or the troll. In several trials, the agent was unable to extricate itself from these situations without being reduced to zero damage, which results in death for the agent. In V-World, damage is accumulated when the agent occupies spaces immediately adjacent to hostile animate actors, or from pushing on them by attempting to occupy the square in which they are located. This problem tends to occur in areas of the world three or fewer squares in one dimension, with a way out that is one square across. Typically, the agent avoids the proximity of hostile actors unless it carries the means to defeat them, such as the sword in the case of the troll or the bane in the case of the dragon. However, these actors are often not visible to the agent upon entry into one of these confined spaces. This is particularly problematic when the agent encounters the dragon, since one turn spent in a space adjacent to the dragon removes fifty damage points, out of a possible maximum of one hundred. The agent has several times encountered the situation of having been damaged by the dragon, with no way out that does not take it past the dragon again. Similar problems have occurred less frequently with the troll, due to the troll's smaller damaging effect.

One solution that was attempted was the implementation of negative spreading activation from the location from which a hostile actor had been observed. The intended effect of this strategy was to cancel out the positive goal influence emitted in the agent's decision layer by objects or unexplored spaces near a threat, influencing the agent away from dangerous areas until it had the means to combat them. Unfortunately, even small quantities of negative activation emitted per turn were sufficient to flood the decision layer and warn the agent away from that entire half of the world. Since animate objects are not represented by the mapping behaviors, the source of negative activation was the location from which the agent first saw the hostile actor, which made focusing the effect on a small geographical area impossible. At present there does not seem to be a way to mark certain areas of unexplored territory as dangerous without resorting to explicit symbolic labeling, since all move behaviors representing new terrain are essentially the same to the agent.

The use of move behaviors that terminated in the actors themselves was also attempted. These move behaviors did not tend to spread exploration activation, which kept the agent from venturing into these areas until the necessary counter-measures were in its possession. The problem with this solution arose when the threat was defeated. The agent is not currently equipped to recognize the persistence of objects in the world. It is therefore difficult for the agent to tell when a hostile actor has been defeated, as

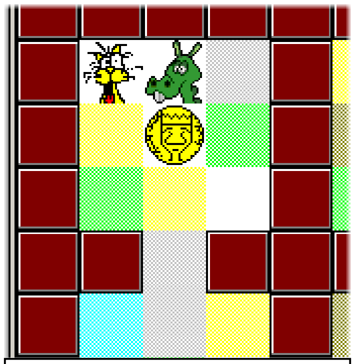


Figure 9: The agent is trapped by the dragon

opposed to when it has moved out of sight. There does not seem to be an appropriate trigger that the agent can recognize for changing the threat-related move behaviors into normal move behaviors, and any other means would violate the intent of the situated design strategy employed. There may be a way of introducing qualitative connections between position and move behaviors in a given area and the types of objects that inhabit that area. It should also be possible to give the agent the concept of persistence, to allow it to handle objects and actors in a manner more resembling that of living organisms. Within the scope of the current project, however, this remains a problem, if an infrequent one. Difficulty handling hostile actors does not signal a failure of the ideas manifested in this agent, but it does suggest areas for improvement of the design.

4.4 CONCLUSION

The agent successfully completes the primary goal of the V-World planning domain, that of returning the princess to the throne. In the process of doing so, it effectively completes sub-goals as they become apparent, in the order required for the fulfillment of the central goal. It is able to reliably navigate the geography of the domain, employing a "good enough" strategy to recognize familiar locations and to determine the position of various locations relative to each other in order to locate and travel to necessary resources and entities. The agent tends to run slowly, but this is not detrimental to its effectiveness in the arena of goal fulfillment. The agent's tendency to oscillate between goals could conceivably present a problem in conditions involving extreme hunger, extreme damage, or proximity to a threat, but has not itself resulted in the agent's failure to date. The problem of becoming stuck in threatened spaces has resulted in several unsuccessful trials, but this occurs infrequently, and offers several possible solutions.

The agent represents an attempt to solve a problem designed for agents based on the Physical Symbol System Hypothesis, and to display greater opportunism, adaptability, flexibility, and fault tolerance while doing so. The agent is able to fulfill the domain requirements, and does display the desired characteristics. The effectiveness of this agent suggests that the principles of situated cognition have much to offer the field of Artificial Intelligence, particularly in the domain of autonomous rational agents. The ideas presented here may also have significant ramifications for other domains within Artificial Intelligence, such as the design of intelligent control systems. The concepts of distributed control systems that do not fail all at once, separate implicit representation schemes for separate tasks that need not all speak the same symbolic language, and modes of perception that do not impose artificial boundaries of abstraction on the external environment all have important ramifications for practical concerns such as robotics or the design of autonomous vehicle control systems. The patterns of behavior displayed by the agent also suggest that this approach to agent design might be a useful direction in which to conduct research as to how living organisms think and act. Such a research direction may discover that a model of natural cognition based on task decomposition and decentralized representation and control is more effective for understanding and duplicating natural intelligence than previous attempts based upon symbol ontologies.

BIBLIOGRAPHY

Brooks, R. (1990) Elephants Don't Play Chess. *Robotics and Autonomous Systems 6*, 3-15.

Brooks, R. (1991) Intelligence Without Reason. *Artificial Intelligence Journal* (47), 1991, pp. 139–159.

Brooks, R. (1997) From Earwigs to Humans. *Robotics and Autonomous Systems 20*, 291–304.

Bridger, B., Crouch, D., and Nute, D. (2000) Planning Agent Architecture for the Virtual World Environment. Proceedings, IC-AI'01.

Maes, P. (1990) Situated Agents Can Have Goals. In Maes, P. eds. *Designing Autonomous Agents* 49-70.

Malcom, C. and Smithers, T. (1990) Symbol Grounding via a Hybrid Architecture in an Autonomous Assembly System. In Maes, P. eds. *Designing Autonomous Agents* 123-144.

Mataric, M. (1990) "A Distributed Model for Mobile Robot Environment-Learning and Navigation", MIT EECS Master's Thesis, Jan 1990.

Mataric, M. (1997) "Situated Robotics". invited contribution to the *Encyclopedia of Cognitive Science,* Nature Publishing Group, Macmillan Reference Limited, to appear in Nov 2002.

Newell, A. and Simon, H. (1976) Computer Science as Empirical Inquiry: Symbols and Search. In Haugeland, J. eds. *Mind Design II* 81-110. Cambridge: MIT Press.