Elephants Don't Play Chess

Individual sections are marked as "Read" or "Skim"

Rodney A. Brooks
MIT Artificial Intelligence Laboratory, Cambridge, MA 02139, USA

Robotics and Autonomous Systems 6 (1990) 3-15

Keywords: Situated activity; Mobile robots; Planning; Subsumption architecture; Artificial Intelligence.

Rodney A. Brooks was born in Adelaide, Australia. He studied Mathematics at the Flinders University of South Australia and received a Ph.D. from Stanford in Computer Science in 1981. Since then he has held research associate positions at Carnegie Mellon University and the Massachusetts Institute of Technology and faculty positions at Stanford and M.I.T. He is currently an Associate Professor of Electrical Engineering and Computer Science at M.I.T. and a member of the Artificial Intelligence Laboratory where he leads the mobile robot group. He has authored two books, numerous scientific papers, and is the editor of the *International Journal of Computer Vision.*

There is an alternative route to Artificial Intelligence that diverges from the directions pursued under that banner for the last thirty some years. The traditional approach has emphasized the abstract manipulation of symbols, whose grounding, in physical reality has . rarely been achieved. We explore a research methodology which emphasizes ongoing physical interaction with the environment as the primary source of constraint on the design of intelligent systems. We show how this methodology has recently had significant successes on a par with the most successful classical efforts. We outline plausible future work along these lines which can lead to vastly more ambitious systems.

Read

1. Introduction

Artificial Intelligence research has foundered in a sea of incrementalism. No one is quite sure where to go save improving on earlier demonstrations of techniques in symbolic manipulation of ungrounded representations. At the same time, small AI companies are folding, and attendance is well down at national and international Artificial Intelligence conferences. While it is true that the use of AI is prospering in many large companies, it is primarily through the application. to novel domains of long developed techniques that have become passé in the research community

What has gone wrong? (And how is this book the answer?!!)

In this paper we argue that the *symbol system hypothesis* upon which *classical AI is* base is fundamentally flawed, and as such imposes severe limitations on the fitness of its progeny. Further, we argue that the dogma of the symbol system hypothesis implicitly includes a number of largely unfounded great leaps of faith when called upon to provide a plausible path to the digital equivalent of human level intelligence. It is the chasms to be crossed by these leaps which now impede classical AI research.

But there is an alternative view, or dogma, variously called *nouvelle AI, fundamentalist AI*, or in a weaker form *situated activity* [1]. *It is based on the physical grounding hypothesis.* It provides a different methodology for building intelligent systems than that pursued for the last thirty years. The traditional methodology bases its decomposition of intelligence into functional information processing modules whose combinations provide overall system behavior. The new methodology bases its decomposition of intelligence into individual behavior generating modules, whose coexistence and co-operation let more complex behaviors emerge.

In classical AI, none of the modules themselves generate the behavior of the total system. Indeed it is necessary to combine together many of the modules to get any behavior at all from the system. Improvement in the competence of the system proceeds by improving the individual functional modules. In nouvelle AI each module, itself generates behavior, and improvement in the competence of the

[1] Note that what is discussed in this paper is completely unrelated to what is popularly known as *Neural Networks.* That given, there are nevertheless a number of aspects of nouvelle AI approaches which may be of interest to. people working in classical neuroscience.

system proceeds by adding new modules to the system.

Given that neither classical nor nouvelle AI seem close to revealing the secrets of the holy grail of AI, namely general purpose human level intelligence equivalence, there are a number of critical comparisons that can be made between the two approaches.

- Is either approach epistemologically adequate? (And adequate for what?)

- Are there clear paths for either approach in the direction of vastly more intelligent systems?

- Are nouvellers romantically hoping for magic from nothing while classicists are willing to tell their systems almost anything and everything, in the hope of teasing out the shallowest of inferences?

- Is the claim of emergent properties of nouvelle AI systems any more outrageous than the use of heuristics in classical AI?

In the following sections we address these issues.

Read

2. The Symbol System Hypothesis

The symbol system hypothesis, [30], states that intelligence operates on a system of symbols. The implicit idea is that perception and motor interfaces are sets of symbols on which the central intelligence system operates. Thus, the central system, or reasoning engine, operates in a domain independent way on the symbols. Their meanings are unimportant to the reasoner, but the coherence of the complete process emerges when an observer of the system knows the groundings of the symbols within his or her own experience.

Somewhat more implicitly in the work that the symbol system hypothesis has inspired, the symbols represent entities in the world. They may be individual objects, properties, concepts, desires, emotions, nations, colors, libraries, or molecules, but they are necessarily named entities. There are a number of effects which result from this commitment.

Recall first, however, that an intelligent system, apart from those which are experiments in the laboratory, will be embedded in the world in some form or another.

2.1. The Interface Between Perception and Symbols

The central intelligence system deals in symbols. It must be fed symbols by, the perception system.

But what is the correct symbolic description of the world around the intelligence system? Surely that description must be task dependent.

The default assumption has been that the perception system delivers a description of the world in terms of typed, named individuals and their relationships. For instance in the classic monkeys and bananas problem, the world, description is in terms of boxes, bananas, and aboveness.

But for another task (e.g., deciding whether the bananas are rotten) quite a different representation might be important. Psychophysical evidence [32] certainly points to perception being an active and task dependent operation.

The effect of the symbol system hypothesis has been to encourage vision researchers to quest after the goal of a general purpose vision system which delivers complete descriptions of the world in a symbolic form (e.g. [5]). Only recently has there been a movement towards active vision [4] which is much more task dependent, or task driven [1].

2.2. Inadequacy of Simple Symbols

Symbol systems in their purest forms assume a knowable objective truth. It is only with much complexity that modal logics, or non-monotonic logics, can be built which better enable a system to have, beliefs gleaned from partial views of a chaotic world.

As these enhancements are made, the realization of computations based on these formal systems becomes more and more biologically implausible. But once the commitment to symbol systems has been made it is imperative to push on through more and more complex and cumbersome systems in pursuit of objectivity.

This same pursuit leads to the well known frame problem (e.g., [27]), where it is impossible to assume anything that is not explicitly stated. Technical deviations around this problem have been suggested but they are by no means without their own problems.

2.3. Symbol Systems Rely on Emergent Properties

In general the reasoning process becomes trivial in an NP-complete space (e.g., There have been large efforts to overcome these problems by choosing simple arithmetically computed *evaluation functions* or polynomials to guide the search. Charmingly, it has been hoped that intelligence will somehow emerge from these simple numeric computations carried out in the sea of symbols. [28] was one of the earliest examples of this hope, which later turned out to be only partially correct (his learned polynomials later turned out to be dominated by piece count), but in fact almost all instances of search in classical AI have relied on such judiciously chosen polynomials to keep the search space manageable.

Read

3. The Physical Grounding Hypothesis

Nouvelle AI is based on the physical grounding hypothesis. This hypothesis states that to build a system that is intelligent it is necessary to have its representations grounded in the physical world. Our experience with this approach is that once this commitment is made, the need for traditional symbolic representations soon fades entirely. The key observation is that the world is its own best model. It is always exactly up to date. It always contains every detail there is to be known. The trick is to sense it appropriately and often enough.

To build a system based on the physical grounding hypothesis it is necessary to connect it to the world via a set of sensors and actuators. Typed input and output are no longer of interest. They are not physically grounded.

Accepting the physical grounding hypothesis as a basis for research entails building systems in a bottom up manner. High level abstractions have to be made concrete. The constructed system eventually has to express all its goals and desires as physical action, and must extract all its knowledge from physical sensors. Thus the designer of the system is forced to make everything explicit. Every short-cut taken has a direct impact upon system competence, as there is no slack in the input/output representations. The forms of the low-level interfaces have consequences which ripple through the entire system.

3.1. Evolution

We already have an existence proof of the possibility of intelligent entities — human beings. Additionally many animals are intelligent to some degree. (This is a subject of intense debate, much of which really centers around a definition of intelligence.) They have evolved over the 4.6 billion year history of the earth.

It is instructive to reflect on the way in which earth-based biological evolution spent its time. Single cell entities arose out of the primordial soup roughly 3.5 billion years ago. A billion years passed before photosynthetic plants appeared. After almost another billion and a half years, around 550 million years ago, the first fish and vertebrates arrived, and then insects 450 million years ago. Then things started moving fast. Reptiles arrived 370 million years ago, followed by dinosaurs at 330 and mammals at 250 million years ago. The first primates appeared 120 million years ago and the immediate predecessors to the great apes a mere 18 million years ago. Man arrived in roughly his present form 2.5 million years ago. He invented agriculture a mere 19000 years ago, writing less than 5000 years ago and "expert" knowledge only over the last few hundred years.

This suggests that problem solving behavior, language, expert knowledge and application, and reason, are all rather simple once the essence of being and reacting are available. That essence is the ability to move around in a dynamic environment, sensing the surroundings to a degree sufficient to achieve the necessary maintenance of life and reproduction. This part of intelligence is where evolution has concentrated its time—it is much harder. This is the physically grounded part of animal systems.

An alternative argument to the preceeding is that in fact once evolution had symbols and representations things started moving rather quickly. Thus symbols are the key invention and AI workers can sidestep the early morass and start working directly with symbols. But I think this misses a critical point, as is shown by the relatively weaker performance of symbol based mobile robots as opposed to physically grounded robots. Without a carefully built physical grounding any symbolic representation will be mismatched to its sensors and actuators. These groundings provide the constraints on symbols necessary for them to be truly useful.

[26] has argued rather eloquently that mobility, acute vision and the ability to carry out survival related tasks in a dynamic environment provide a necessary basis for the development of true intelligence.

3.2. The Subsumption Architecture

Read

In order to explore the construction of physically grounded systems we have developed a computational architecture known as the *subsumption architecture*. It enables us to tightly connect perception to action, embedding robots concretely in the world.

A subsumption program is built on a computational substrate that is organized into a series of incremental layers, each, in the general case, connecting perception to action. In our case the substrate is, networks of finite state machines augmented with timing elements.

The subsumption architecture was described initially in [6] and later modified in [8] and [16]. The subsumption compiler compiles augmented finite state machine (AFSM) descriptions into a special-purpose scheduler to simulate parallelism and a set of finite state machine simulation routines. This is a dynamically retargetable compiler that has backends for a number of processors, including the Motorola 68000, the Motorola 68HC11, and the Hitachi 6301. The subsumption compiler takes a source file as input and produces an assembly language program as output.

The behavior language was inspired by [23] as a way of grouping AFSMs into more manageable units with the capability for whole units being selectively activated or de-activated. In fact, AFSMs are not specified directly, but rather as rule sets of real-time rules which compile into AFSMs in a one-to-one manner. The behavior compiler is machine-independent and compiles into an intermediate file of subsumption AFSM specifications. The subsumption compiler can then be used to compile to the various targets., We sometimes call the behavior language the *new subsumption.*

Skim

3.2.1. The Old Subsumption Language

Each augmented finite state machine (AFSM) has a set of registers and a set of timers, or alarm clocks, connected to a conventional finite state machine which can control a combinational network fed by the registers. Registers can be written by attaching input wires to them, and sending messages from other machines. The messages get written into the registers by replacing any existing contents. The arrival of a message, or the expiration of a timer, can trigger a change of state in the interior finite state machine. Finite state machine states can either wait on some event, conditionally dispatch to one of two other states based on some combinational predicate on the registers, or compute a combinational function of the registers directing the result either back to one of the registers or to an output of the augmented finite state machine. Some AFSMs connect directly to robot hardware. Sensors deposit their values in certain registers, and certain outputs direct commands to actuators.

A series of layers of such machines can be augmented by adding new machines and connecting them into the existing network in a number of ways. New inputs can be connected to existing registers, which might previously have contained a constant. New machines can inhibit existing outputs, or suppress existing inputs, by being attached as side-taps to existing wires. When a message arrives on an inhibitory side-tap no messages can travel along the existing wire for some short time period. To maintain inhibition there must be a continuous flow of messages along the new wire. (In previous versions of the subsumption architecture [6] explicit, long time periods had to be specified for inhibition or suppression with single shot messages., Recent work has suggested this better approach [16].) When a message arrives on a suppressing side-tap, again no messages are allowed to flow, from the original source for some small time period, but now the suppressing message is gated through and it masquerades as having come from the original source. A continuous supply of suppressing messages is required to maintain control of a side-tapped wire.

Inhibition and suppression are the mechanisms by which conflict resolution between actuator commands from different layers is achieved. Notice that in this definition of the subsumption architecture, AFSMs cannot share any state, and in particular they each completely encapsulate their own registers and alarm clocks.

All clocks in a subsumption system have approximately the same tick period (0.04 seconds in most of our robots). However, neither the clocks nor the messages are synchronous. The fastest possible rate of sending messages along a wire is one per clock tick. The time periods used for both inhibition and suppression are two clock ticks. Thus, a side-tapping wire with messages being sent at the maximum rate can maintain control of its host wire. We call this rate the *characteristic frequency* of the particular subsumption implementation.

Skim

3.2.2. The New Subsumption Language

The behavior language groups multiple processes (each of which usually turns out to be implemented as a single AFSM) into *behaviors.* There can be message passing, suppression, and inhibition between processes within a behavior, and there can be message passing, suppression and inhibition between behaviors. Behaviors act as abstraction barriers; one behavior cannot reach inside another.

Each process within a behavior is much like an AFSM, and indeed our compiler for the behavior language converts them to AFSMs. However, they are generalized so that they can share registers. A new structure, monostables, provides a slightly more general timing mechanism than the original alarm clocks. Monostables are retriggerable, and can be shared between processes within a single behavior.

See below

4. Some Physically Grounded Systems

In this section we briefly review some previous successful robots built with the subsumption architecture and highlight the ways in which they have exploited or epitomize that architecture. The family portrait of all the robots is shown in Fig. 1. Most of the robots were programmed with the old subsumption language. Toto and Seymour use the new behavior language.

A key thing to note with these robots is the ways in which seemingly goal-directed behavior emerges from the interactions of simpler non goal-directed behaviors.

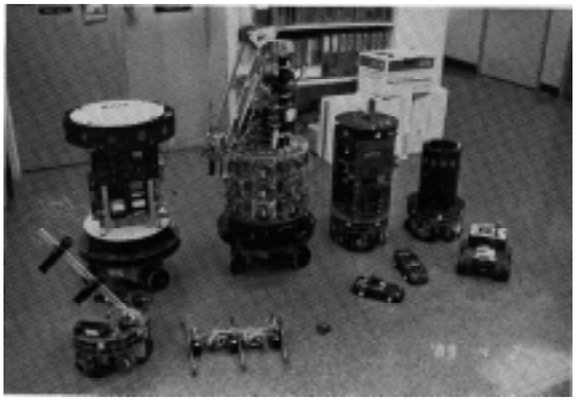

Fig. 1. The MIT Mobile Robots include, in the back row, left to right; Allen, Herbert, Seymour and Toto. In front row are Tito, Genghis, Squirt (very small) Tom and Jerry, and Labnav.

Read

4.1. Allen

Our first robot, Allen, had sonar range sensors and odometry onboard and used an offboard lisp machine to simulate the subsumption architecture. In [6] we described three layers of control implemented in the subsumption architecture.

The first layer let the robot avoid both static and dynamic obstacles; Allen would happily sit in the middle of a room until approached, then scurry away, avoiding collisions as it went. The internal representation used was that every sonar return represented a repulsive force with an inverse square decrease in strength as a function of distance. The vector sum of the repulsive forces, suitably thresholded, told the robot in which direction it should move. An additional reflex halted the robot whenever there was something right in front of the robot and it was moving forward (rather than turning in place).

The second layer made the robot randomly wander about. Every 10 seconds or so, a desire to head in a random direction would be generated. That desire was coupled with the reflex to avoid obstacles by vector addition. The summed vector suppressed the more primitive obstacle avoidance vector, but the obstacle avoidance behavior still operated, having been subsumed by the new layer, in its account of the lower level's repulsive force. Additionally, the halt reflex of the lower level operated autonomously and unchanged.

The third layer made the robot look (with its sonars) for distant places and try to head towards them. This layer monitored progress through odometry, generating a desired heading which suppressed the direction desired by the wander layer. The desired heading was then fed, into a vector addition with the instinctive obstacle avoidance layer. The physical robot did not therefore remain true to the desires of the upper layer. The upper layer had to watch what happened in the world, through odometry, in order to understand what was really happening in the lower control layers, and send down correction signals.

In [9] we described an alternate set of layers for the robot Allen.

Read

4.2. Tom and Jerry

Tom and Jerry [14] were two identical robots built to demonstrate just how little raw computation is necessary to support the subsumption architecture. A three layer subsumption program was implemented, yet all data paths were just one bit wide and the whole program fitted on a single 256 gate programmable array logic chip. Physically Tom and Jerry were toy cars with three one-bit infrared proximity sensors mounted on the front and one at the rear. The sensors were individually tuned to a specific distance at which they would fire. The central front sensor fired only on much closer objects than the two side sensors, which pointed slightly outward.

The lowest layer of Tom and Jerry implemented the standard pair of first level behaviors. These used a vector sum of repulsive forces from obstacles to perform an avoidance manuever or to trigger a halt reflex to stop when something was too close ahead, as detected, by the central front looking sensor. There were extra complications with Tom and Jerry in that we needed to use the subsumption architecture to implement an active braking scheme because of the high speed of the robots relative to their sensor ranges. Tom and Jerry's second layers were much like Allen's original second layer-an urge to wander about, which was implemented by an attractive force which got added to the repulsive forces from obstacles. The third layer detected moving objects using the front three sensors and created a following behavior. When something was detected, the robot was attracted and moved towards it. The lower level collide behavior stopped the robot from actually hitting the target, however. While the robot was chasing its target, the wander behavior was suppressed.

Tom and Jerry demonstrated the notion of independent behaviors combining without knowing about each other (chasing obstacles but staying back from them a little). Tom and Jerry also demonstrated that the subsumption architecture could be compiled (by hand) down to the gate level, and that it could be run at clock speeds of only a few hundred Hertz.

Read

4.3. Herbert

Herbert [12] was a much more ambitious robot. It has a 24-processor distributed, loosely coupled, onboard computer to run the subsumption architecture. The processors were slow CMOS 8-bit microprocessors (which ran on low electrical power; an important consideration when carrying batteries), which could communicate only by slow serial interfaces (maximum 10 packets each, 24 bits wide per second). Onboard Herbert, the interconnections between AFSMs are physically embodied as actual copper wires.

Herbert had 30 infrared proximity sensors for local obstacle avoidance, an onboard manipulator with a number of simple sensors attached to the hand, and a laser light striping system to collect three dimensional depth data in a 60 degree wide swath in front of the robot with a range of about 12 feet. A 256 pixel-wide by 32 pixel-high depth image was collected every second. Through a special purpose distributed serpentine memory, four of the onboard 8-bit processors were each able to expend about 30 instructions on each data pixel. By linking the processors in a chain we were able to implement quite high performance vision algorithms.

[16] programmed Herbert to wander around office areas, go into people's offices and steal empty soda cans from their desks. He demonstrated obstacle avoidance and wall following, real-time recognition of soda-can-like objects, and a set of 15 behaviors [15] which drove the arm to physically search for a soda can in front of the robot, locate it, and pick it up.

Herbert showed many instances of using the world as its own best model and as a communication medium. The remarkable thing about Herbert is that there was absolutely no internal communication between any of its behavior generating modules. Each one was connected to sensors on the input side, and an arbitration network on the output side. The arbitration network drove the actuators.

The laser-based soda-can object finder drove the robot so that its arm was lined up in front of the soda can. But it did not tell the arm controller that there was now a soda can ready to be picked up. Rather, the arm behaviors monitored the shaft encoders on the wheels, and when they noticed that there was no body motion, initiated motions of the arm, which in turn triggered other behaviors, so that eventually the robot would pick up the soda can.

The advantage of this approach is that there is no need to set up internal expectations for what is going to happen next; this means that the control system can both (1) be naturally opportunistic if fortuitous circumstances present themselves, and (2) it can easily respond to changed circumstances, such as some other object approaching it on a collision course.

As one example of how the arm behaviors cascaded upon one another, consider actually grasping a soda can. The hand had a grasp reflex that operated whenever something broke an infrared beam between the fingers. When the arm located a soda can with its local sensors, it simply drove the hand so that the two fingers lined up on either side of the can. The hand then independently grasped the can. Given this arrangement, it was possible for a human to hand a soda can to the robot. As soon as it was grasped, the arm retracted — it did not matter whether it was a soda can that was intentionally grasped, or one that magically appeared. The same opportunism among behaviors let the arm adapt automatically to a wide variety of cluttered desktops, and still successfully find the soda can.

4.4. Genghis

Skim

Genghis [81 is a 1Kg six legged robot which walks under subsumption control and has an extremely distributed control system. The robot successfully walks over rough terrain using 12 motors, 12 force sensors, 6 pyroelectric sensors, one inclinometer and 2 whiskers. It also follows cooperative humans using its pyroelectric sensors.

The subsumption layers successively enable the robot to stand up, walk without any sensing, use force measurements to comply with rough terrain, use force measurements to lift its legs over obstacles, use inclinometer measurements to selectively inhibit rough terrain compliance when appropriate, use whiskers to lift feet over obstacles, use passive infrared sensors to detect people and to walk only when they are present, and to use the directionality of infrared radiation to modulate the backswing of particular leg sets so, that the robot follows a moving source of radiation.

In contrast, one could imagine a control system which had a central repository which modeled the robot's configuration in translation and orientation space. One could further imagine high level commands (for instance from a path planner) generating updates for the robot's coordinates. These high level commands would then be hierarchically resolved into instructions for individual legs.

The control system on Genghis has no such repository. Indeed there is not even a central repository for each leg—separate motors on the legs are controlled quite separately in different parts of the network. While there is a some semblance of a central control system for each individual motor, these controllers receive messages from diverse parts of the network and simply pass them on to the motors, without any attempt at integration.

Our control system was also very easy to build. It was built incrementally, with each new capability being a simple addition (no deletion, no change to previous network) of new network structure. The debugged existing network structure was never altered.

The resulting control system is elegant in its simplicity. It does not deal with coordinate transforms or kinematic models. It is not at all hierarchical. It directly implements walking through many very tight couplings of sensors to actuators. It is very distributed in its nature, and we believe its robustness in handling rough terrain comes from this distributed form of control.

We are currently building a new version of Genghis [3] which will be a much stronger climber and able to scramble at around three kilometers per hour. Each leg has three degrees of freedom and three force sensors mounted on load bearing beams. A single-chip microprocessor with onboard RAM and EEPROM is easily able to force servo the complete leg. The total mass of the final robot will be 1.6 Kg. Attila will have batteries which will power it for about 30 minutes while actively walking. Following that, it will have to recharge from solar cells for about 4.5 hours in Earth sunlight.

Skim

4.5. Squirt

Squirt is the smallest robot we have built [21]. It weighs about 50 grams and is about 5/4 cubic inches in volume.

Squirt incorporates an 8-bit computer, an onboard power supply, three sensors and a propulsion system. Its normal mode of operation is to act as a "bug", biding in dark corners and venturing out in the direction of noises, only after the noises are long gone, looking for a new place to hide near where the previous set of noises came from.

The most interesting thing about Squirt is the way in which this high level behavior emerges from a set of simple interactions with the world.

Squirt's lowest level of behavior monitors a light sensor and causes it to move in a spiral pattern searching for darkness. The spiral trajectories are created by a coupling of a forward motion along with a back-and-turn motion, implemented through the use of only one motor and made possible by a unidirectional clutch on the rear axle. Once Squirt finds a dark spot, it stops.

Squirt's second level of behavior is triggered once a dark hiding place has been established. This behavior monitors two microphones and measures the time of arrival of sound at each microphone. By noting the difference, it can localize the direction from which the sound came. Squirt then waits for a pattern of a sharp noise followed by a few minutes of silence. If this pattern is recognized, Squirt ventures out in the direction of the last heard noise, suppressing the desire to stay in the dark. After this ballistic straight-line motion times out, the lower level is no longer suppressed and the light sensor is again recognized. If it is light, the spiraling pattern kicks back in. The end effect is that Squirt gravitates towards the center of action. The entire compiled control system for Squirt fits in 1300 bytes of code on an onboard microprocessor.

4. 6 Toto

Skim

Toto [24] is our first robot fully programmed with the new behavior language. Toto has 12 radially arranged sonars and a flux-gate compass as its sensors.

At first appearance it may seem that the subsumption architecture does not allow for such conventional items as maps. There are no data structures within the subsumption architecture, and no easy way of having a central repository for more than simple numeric quantities. Our work with Toto demonstrates that these are not critical limitations with regard to map building and use.

Toto has a low level reactive system to keep basic functions running robustly. Its lower level behaviors enable it to wander around avoiding collisions, and successfully follow walls and corridors as if it were explicitly exploring the world. An intermediate level set of behaviors tries to recognize particular types of landmark such as walls, corridors and clutter. Another network is made up of mutually identical behaviors with each layer waiting for new landmarks to be recognized. Each time this happens a behavior allocates itself to be the 'place' of that particular landmark. The behaviors which correspond to physically adjacent landmarks have neighbor relationship links activated between them. A graph structure is thus formed, although the nodes are active computational elements rather than static data structures. (In fact, each node is really a whole collection of computational elements in the form of augmented finite state machines.)

As the robot moves around the environment, the nodes try to keep track of where it is. Nodes become more active if they believe that they correspond to the place at which the robot is currently located. Thus the robot has both a map, and a sense of where it is on the map, but a totally distributed computational model.

When a behavior (such as "go to some place") is activated (via a small panel of push buttons on the robot) a spreading of activation mechanism is used, which spreads from the goal via the neighbor links. This process is continuous and keeps the robot informed as it reaches each place expected from the map.

Mataric's experimental results [25] show how the robot's performance can be incrementally improved by adding new pieces of network. Map building and path planning were initially demonstrated with fewer types of behaviors than finally implemented. Then an idea of expectation, based on temporally generated context was added. This allowed the robot to handle getting lost and to relocate itself in the map later. Then a coarse position estimation scheme was added, based on integrating the compass heading over time. This significantly lowered the level of ambiguity in both map building and map use in more complex environments, and thus increased the robot's overall competence. In all cases we simply added new behaviors to the network to improve the map building and using performance.

The work has also shown that globally consistent maps can be built and emerge in a totally distributed manner. In our experiments they were built by a collection of asynchronous independent agents, without the ability to use arbitrary pointers, or other such traditional data structure techniques. In path planning there is no notion of a global path under this scheme; local pieces of information combine to direct the robot through its dynamics of interaction with the world, to get to the desired place. Overall, these aspects demonstrate that the techniques should scale well.

It has been easy to integrate the maps with the dynamics of navigation, obstacle avoidance and path planning. The representations have a natural ability to integrate temporal aspects of the dynamics since they can use time as its own representation!

The notion of place maps developed for Toto bears striking similarities to what has been observed in the hippocampus of the rat [17].

Skim

4.7. Seymour

Seymour is a new robot we are building with all onboard processing to support vision processing of 9 low resolution cameras at approximately 10 frames per second [10]. The cameras feed into different subsumption layers which act upon those aspects of the world they perceive. Seymour is also programmed in the new behavior language.

A number of vision based behaviors developed for Seymour have been prototyped on earlier robots.

[22] describe a subsumption program that controls two simple and unreliable visual processing routines to produce a reliable behavior which follows moving objects using vision. One vision process tracks a single moving blob. It gets bootstrapped by another process which overlays the blob image with an indication of where motion is seen. The robot then tries to servo a selected blob to stay in a fixed location in image coordinates. The blob tracker often loses the blob it is tracking. The motion finder produces a lot of noise especially when the robot is moving, but between the two of them they let the robot reliably follow a moving object (any moving object; we have seen the robot chase a black trash can dragged by a string, a radio controlled blue toy car on a blue floor, a pink plastic flamingo, a grey notebook on a grey carpeted floor, and a drinking mug moved around by hand), by switching back, and forth between the visual routines as either one fails. Nowhere internally does the subsumption program have the notion of an identifiable object, yet to an outside observer it certainly appears to follow a moving object very well.

Using the robot Tito, [29] demonstrated two visually guided behaviors which will be used in support of Seymour. Each behavior used a stereo pair of linear cameras. A vertically mounted pair made use of rotational motions of the base to produce images from which the dimensions of the room could be extracted even though the camera system was uncalibrated. Then employing earlier results from [11], the robot used forward motion to calibrate a horizontally mounted pair of cameras, which were used to find doorways through which the robot drove.

[31] has demonstrated an autonomous eyeball capable of maintaining a steady gaze despite motion of its platform. It recapitulates the primate vestibular-occular system by using vision as a slow calibration system for a gyroscope controlled movable platform which holds the camera.

Skim

4.8. Gnat Robots

In all our use and development of the subsumption architecture we have been careful to maintain its simplicity so that programs written in it could be easily and mechanically compiled into silicon. For example, with Toto the map networks were arranged so that the total wire length for connecting the underlying finite state machines need be no more than linear in the number of finite state machines. In general the area of silicon needed for the robots we have built would be quite small. There is a reason for maintaining this restriction.

[18,19] introduced the idea of building complete small robots out of silicon on a VLSI fabrication line. [7] demonstrated how to use the subsumption architecture to control such robots. There is great potential for using such robots in ways previously not considered at all cost effective for robotic applications. Imagine, for instance having a colony of tiny robots living on your TV screen, absorbing energy from the electron beam, whose only purpose in existence is to keep the screen clean. There is potential for a revolution in micro-mechanical systems of the same order and impact as the quiet revolutions brought about in daily life by the advent of the micro-processor.

[20] outlines a series of technological steps necessary to build such robots, including materials, a new type of micro motor based on thin film piezo-electric material, a 3-D fabrication process, and some new types of integrated sensors. Critical to this enterprise is an easy way of controlling the robots, giving them intelligent behavior in unstructured and uncertain environments.

Read

5. Measures of Success

When I give talks about the techniques we have used to build intelligent control systems for our robots, the most common questions I am asked, or assertions I am told, are:

- “If I make *such-and-such* a change to your robot's environment, I bet it would do the wrong thing.”

- “Aren't these systems almost impossible to debug?”

- “Surely this can't be scaled up to do X,” for some value of X which has not been part of the talk.

In the next three subsections I argue that these questions are either easy to answer or, in a deep sense, improper to ask.

5. 1. Puzzlitis

Since traditional Artificial Intelligence research has concentrated on isolated modules of intelligence that almost never get grounded in the world, it has been important to develop some criteria for successful research. One of the most popular ideas is generality. This quickly leads to a disease I call *puzzlitis.* The way to show generality is to pick the most obscure case within the domain and demonstrate that your system can handle or solve it.

But in physically grounded systems I believe this approach is counterproductive. The puzzles posed are often very unlikely in practice, but to solve them makes the systems much more complex. This reduces the overall robustness of the system! We should be driven by puzzles which can naturally arise in a physically grounded context—this is what gives strength to our physically grounded systems.

One additional argument on this topic is that for most AI programs the creator gets to tell the program the facts in some sort of representation language. It is assumed that the vision guys in the white hats down the corridor will one day deliver world models using these same representations. Many of the puzzlitis failures of physically grounded systems stem from a failure in perception as the stakes have been raised. Standard AI programs have not been forced to face these issues.

5.2. Debugging

In our experience debugging the subsumption programs used to control our physically grounded systems has not been a great source of frustration or difficulty. This is not due to any particularly helpful debugging tools or any natural superiority of the subsumption architecture.

Rather, we believe it is true because the world is its own best model (as usual). When running a physically grounded system in the real world, one can see at a glance how it is interacting. It is right before your eyes. There are no layers of abstraction to obfuscate the dynamics of the interactions between the system and the world. This is an elegant aspect of physically grounded systems.

5.3. But It Can't Do X

Along with the statement "But it can't do X" there is an implication, sometimes vocalized, and sometimes not, that therefore there are lots of things that this approach is not good for, and so we should resort to the symbol system hypothesis.

But this is a fallacious argument, even if only implicit. We do not usually complain that a medical expert system, or an analogy program cannot climb real mountains. It is clear that their domain of expertise is somewhat more limited, and that their designers were careful to pick a well circumscribed domain in which to work. Likewise it is unfair to claim that an elephant has no intelligence worth studying just because it does not play chess.

People working on physically grounded systems do, however, seem to be claiming to eventually solve the whole problem. E.g., papers such as this one, argue that this is an interesting approach to pursue for precisely that reason. How can we have it both ways?

Like the advocates of the symbol system hypothesis, we believe that in principle we have uncovered the fundamental foundation of intelligence. But just as the symbol system people are allowed to work incrementally in their goals, so should the physical grounding people be allowed. Solutions to all problems are not obvious now. We must spend time, analyzing the needs of certain domains *from the perspective of the physical grounding hypothesis* to discern what new structures and abstractions must be built in order to make forward progress.

Read

6. Future Limits

As [30] points out, concerning his symbol system hypothesis: The hypothesis is clearly an empirical one, to be judged true or false on the basis of evidence. The same can, of course, be said for the physical grounding hypothesis.

Our current strategy is to test the limitations of the physical grounding hypothesis by building robots which are more independent and can do more in the world. We are tackling aspects of human competence in a different order than that chosen by people working under the symbol system hypothesis, so sometimes it is hard to make comparisons between the relative successes. A further part of our strategy then, is to build systems that can be deployed in the real world. At least if our strategy does not convince the arm chair philosophers, our engineering approach will have radically changed the world we live in.

6.1. Contrasts In Hope

Adherents of both approaches to intelligence are relying on some degree of hope that their approach will eventually succeed. They have both demonstrated certain classes of success, but both can resort only to vague hopes when it comes to generalizability. It turns out that the demonstrations and generalization issues fall along different dimensions for the two approaches.

- Traditional AI has tried to demonstrate sophisticated reasoning in rather impoverished domains. The hope is that the ideas used will generalize to robust behavior in more complex domains.

- Nouvelle AI tries to demonstrate less sophisticated tasks operating robustly in noisy complex domains. The hope is that the ideas used will generalize to more sophisticated tasks.

Thus the two approaches appear somewhat complementary. It is worth addressing the question of whether more power may be gotten by combining the two approaches. However, we will not pursue that question further here.

Both approaches rely on some unanalyzed aspects to gain their successes.

Traditional AI relies on the use of heuristics to control search. While much mathematical analysis has been carried out on this topic, the user of a heuristic still relies on an expected distribution of cases within the search tree to get a "reasonable" amount of pruning in order to make the problem manageable.

Nouvelle AI relies on the emergence of more global behavior from the interaction of smaller behavioral units. As with heuristics there is no a priori guarantee that this will always work.. However, careful design of the simple behaviors and their interactions can often produce systems with useful and interesting emergent properties. The user again is relying on expectations without hard proofs.

Can there be a theoretical analysis to decide whether one organization for intelligence is better than another? Perhaps, but I think we are so far away from understanding the correct way of formalizing the dynamics of interaction with the environment that no such theoretical results will be forthcoming in the near term.

6.2. Specific Problems

Some of the specific problems which must be tackled soon, and solved, by approaches to AI based on the physical grounding hypothesis include

- how to combine many (e.g. more than a dozen) behavior generating modules in a way which lets them be productive and cooperative

- how to handle multiple sources of perceptual information when there really does seem to be a need for fusion

- how to automate the building of interaction interfaces between behavior generating modules, so that larger (and hence more competent) systems can be built

- how to automate the construction of individual behavior generating modules, or even to automate their modification

The first two items have specific impact on whether the approach can scale in principle to larger and more complex tasks. The last two are concerned with the issue of how to build such larger systems even if they are in principle possible.

There is room for plenty of experimentation, and eventually, when we are mature enough, there is also room for much theoretical development of the approaches to Artificial Intelligence based on the physical grounding hypothesis.

Acknowledgements

Pattie Maes encouraged me to write this paper, despite my earlier refusal to do so. She and Maja Mataric made a number of useful criticisms of an earlier draft of this paper.

Funding for this work was provided by a number of government agencies and companies including: the University Research Initiative under Office of Naval Research contract N00014-86K-0685, the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-K-0124, Hughes Research Laboratories Artificial Intelligence Center in Malibu, Siemens Research Center in Princeton, and Mazda Research Center in Yokohama.

References

[1] Philip Agre and David Chapman, Pengi: An implementation of a theory of activity, *AAAI-86,* Seattle, WA (1987) 268-272.

[2] Colin M. Angle, Genghis, a six legged autonomous walking robot, MIT S.B. Thesis in Electrical Engineering and Computer Science (March 1989).

[3] Colin M. Angle, Attila's crippled brother marvin, MIT Term Paper in Electrical Engineering and Computer Science (May 1989).

[4] Dana H. Ballard, Reference frame for active vision, IJCAI-89, Detroit, MI (1989) 1635-1641.

[5] Rodney A. Brooks, Symbolic reasoning among 3-D model and 2-D images, *Artificial Intelligence* 17 (1989) 285-348.

[6] Rodney A. Brooks, A robust layered control system for a mobile robot, *IEEE J. Robotics and Automation,* RA-2, April (1986) 14-23.

[7] Rodney A. Brooks, Micro-brains for micro-brawn; Autonomous microbots, *IEEE Micro Robots and Teleoperators Workshop,* Hyannis, MA (November 1987).

[8] Rodney A. Brooks, A robot that walks: Emergent behavior form a carefully evolved network, *Neural Computation,* 1(2) (Summer 1989) 253-262.

[9] Rodney A. Brooks and Jonathan H. Connell, Asynchronous distributed control system for a mobile robot, *SPIE Vol. 727 Mobile Robots,* Cambridge, MA (November 1986) 77-84.

[10] Rodney A. Brooks and Anita M. Flynn, Robot beings, *1EEE/RSI International Workshop on Intelligent Robots and Systems '89,* Tsukuba, Japan (1989) 2-10.

[11] Rodney A. Brooks, Anita M. Flynn, and Thomas Marill, Self calibration of motion and stereo for mobile robot navigation, *MIT AI Memo* 984 (August 1987).

[12] Rodney A. Brooks, Jonathan H. Connell, and Peter Ning, Herbert: A second generation mobile robot, *MIT AI Memo* 1016 (January 1988).

[13] David Chapman, Planning for conjunctive goals, *Artificial Intelligence* 32 (1987) 333-377.

[14] Jonathan H. Connell, Creature building with the subsumption architecture, *IJCAI-87,* Milan (August 1987) 1124-1126.

[15] Jonathan H. Connell, A behavior-based arm controller, *MIT AI Memo* 1025 (June 1988).

[16] Jonathan H. Connell, A colony architecture for an artificial creature, MIT Ph.D. Thesis in Electrical Engineering and Computer Science, MIT AI Lab Tech Report 1151 (June 1989).

[17] H. Eichenbaum, S.I. Wiener, M.L. Shapiro, and N.J. Cohen, The organization of spatial coding in the hippocampus: A study of neural ensemble activity. J. *Neuroscience* 9(8) (1989) 2764-2775.

[18] Anita M. Flynn, Gnat robots (and how they will change robotics), *IEEE Micro Robots and Teleoperators Workshop,* Hyannis, MA (November 1987).

[19] Anita M. Flynn, Gnat robots: A low-intelligence, low-cost approach, *IEEE Solid-State Sensor and Actuator Workshop,* Hilton Head, SC (June 1988).

[20] Anita M. Flynn, Rodney A. Brooks and Lee S. Tavrow, Twilight zones and cornerstones: A gnat robot double feature, *MIT AI Memo 1126* (July 1989).

[21] Anita M. Flynn, Rodney A. Brooks, William M. Wells and David S. Barrett, The world's largest one cubic inch robot, *Proceedings IEEE Micro Electro Mechanical Systems,* Salt Lake City, Utah (February 1989) 98-101.

[22] Ian D. Horswill and Rodney A. Brooks, Situated vision in a dynamic world: Chasing objects, *AAAI-88,* St Paul, MN (August 1988) 796-800.

[23] Pattie Maes, The dynamics of action selection, *ICAI-89,* Detroit, MI (1989) 991-997.

[24] Maja J. Mataric, Qualitative sonar based environment learning for mobile robots, *SPIE Mobile Robots,* Philadelphia, PA (November 1989).

[25] Maja J. Mataric, A model for distributed mobile robot environment learning and navigation, MIT M.S. Thesis in Electrical Engineering and Computer Science (January 1990).

[26] Hans P. Moravec, Locomotion, vision and intelligence, *Robotics Research 1,* Brady and Paul (eds) (MIT Press 1984) 215-224.

[27] Zenon W. Pylyshyn (ed), The Robot's Dilemma, (Ablex Publishing, Norwood, NJ, 1987).

[28] Arthur L. Samuel, Some studies in machine learning using the game of checkers, *IBM.* J. *Res. Development* 3(3) (1959).

[29] Karen B. Sarachik, Characterising an indoor environment with a mobile robot and uncalibrated stereo, *Proceedings IEEE Robotics and Automation,* Scottsdale, AZ (May 1989) 984-989.

[30] Herbert A. Simon, The Sciences of the Artificial (MIT Press, Cambridge, MA, 1969).

[31] Paul Viola, Neurally inspired plasticity in oculomotor processes, *1989 IEEE Conference on Neural Information Processing Systems-Natural and Synthetic,* Denver, CO (November 1989).

[32] A.L. Yarbus, Eye Movements and Vision (Plenum Press 1967).