New Approaches to Robotics

RODNEY A. BROOKS

The author is in the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

In order to build autonomous robots that can carry out useful work in unstructured environments new approaches have been developed to building intelligent systems. The relationship to traditional academic robotics and traditional artificial intelligence is examined. In the new approaches a tight coupling of sensing to action produces architectures for intelligence that are networks of simple computational elements which are quite broad, but not very deep. Recent work within this approach has demonstrated the use of representations, expectations, plans, goals, and learning, but without resorting to the traditional uses, of central, abstractly manipulable or symbolic representations. Perception within these systems is often an active process, and the dynamics of the interactions with the world are extremely important. The question of how to evaluate and compare the new to traditional work still provokes vigorous discussion.

THE FIELD OF ARTIFICIAL INTELLIGENCE (AI) tries to make computers do things that, when done by people, are described as having indicated intelligence. The goal of AI has been characterized as both the construction of useful intelligent systems and the understanding of human intelligence (1). Since AI's earliest days (2) there have been thoughts of building truly intelligent autonomous robots. In academic research circles, work in robotics has influenced work in Al and vice versa (3).

Over the last 7 years a new approach to robotics has been developing in a number of laboratories. Rather than modularize perception, world modeling, planning, and execution, the new approach builds intelligent control systems where many individual modules each directly generate some part of the behavior of the robot. In the purest form of this model each module incorporates its own perceptual, modeling, and planning requirements. An arbitration or mediation scheme, built within the framework of the modules, controls which behavior-producing module has control of which part of the robot at any given time.

The work draws its inspirations from neurobiology, ethology, psychophysics, and sociology. The approach grew out of dissatisfactions with traditional robotics and Al, which seemed unable to deliver real-time performance in a dynamic world. The key idea of the new approach is to advance both robotics and Al by considering the problems of building an autonomous agent that physically is an autonomous mobile robot and that carries out some useful tasks in an environment that has not been specially structured or engineered for it.

There are two subtly different central ideas that are crucial and have led to solutions that use behavior-producing modules:

• *Situatedness:* The robots are situated in the world—they do not deal with abstract descriptions, but with the "here" and "now" of the environment that directly influences the behavior of the system.

• *Embodiment:* The robots have bodies and experience the world directly-their actions are part of a dynamic with the world, and the actions have immediate feedback on the robots' own sensations.

An airline reservation system is situated but it is not embodied—it deals with thousands of request per second, and its responses vary as its database changes, but it interacts with the world only through sending and receiving messages. A current generation industrial spray-painting robot is embodied but it is not situated—it has a physical extent and its servo routines must correct for its interactions with gravity and noise present in the system, but it does not perceive any aspects of the shape of an object presented to it for painting and simply goes through a pre-programmed series of actions.

This new approach to robotics makes claims on how intelligence should be organized that are radically different from the approach assumed by traditional AI.

Traditional Approaches

Although the fields of computer vision, robotics, and AI all have their fairly separate conferences and specialty journals, an implicit intellectual pact between them has developed over the years. None of these fields is experimental science in the sense that chemistry, for example, can be an experimental science. Rather, there are two ways in which the fields proceed. One is through the development and synthesis of models of aspects of perception, intelligence, or action, and the other is through the construction of demonstration systems (4). It is relatively rare for an explicit experiment to be done. Rather, the demonstration systems are used to illustrate a particular model in operation. There is no control experiment to compare against, and very little quantitative data extraction or analysis. The intellectual pact between computer vision, robotics, and AI concerns the assumptions that can be made in building demonstration systems. It establishes conventions for what the components of an eventual fully situated and embodied system can assume about each other. These conventions match those used in two critical projects from 1969 to 1972 which set the tone for the next 20 years of research in computer vision, robotics, and AI.

At the Stanford Research Institute (now SRI International) a mobile robot named Shakey was developed (5). Shakey inhabited a set of specially prepared rooms. It navigated from room to room, trying to satisfy a goal given to it on a teletype. It would, depending on the goal and circumstances, navigate around obstacles consisting of large painted blocks and wedges, push them out of the way, or push them to some desired location. Shakey had an onboard blackand-white television camera as its primary sensor. An offboard computer analyzed the images and merged descriptions of what was seen into an existing , sting symbolic logic model of the world in the form of first order predicate calculus. A planning program, STRIPS, operated on those symbolic descriptions of the world to generate a sequence of actions for Shakey. These plans were translated through a series of refinement into calls to atomic actions in fairly tight feedback loops with atomic sensing operations using Shakey's other sensors, such as a-bump bar and odometry.

Shakey only worked because of very careful engineering of the environment. Twenty years later, no mobile robot has been demonstrated matching all aspects of Shakey's performance in a more general environment, such as an office environment. The rooms in which Shakey operated were bare except for the large colored blocks and wedges. This made the class of objects that had to be represented, very simple. The walls were of a uniform color and carefully lighted, with dark rubber baseboards, making clear boundaries with the lighter colored floor. This, meant that very simple and robust vision of trihedral corners between two walls and the floor could be used for relocalizing the robot in order to correct for drift in the odometric measurements. The blocks and wedges were painted different colors on different planar surfaces. This ensured that it was relatively easy, especially in the good lighting provided, to find edges in the images separating the surfaces and thus to identify the shape of the polyhedron. Blocks and wedges were relatively rare in the environment, eliminating problems due to partial obscurations.

At MIT, a camera system and a robot manipulator arm were programmed to perceive an arrangement of white wooden blocks against a black background and to build a copy of the structure from additional blocks. This was called the *copy-demo (6).* The programs to do this were very specific to the world of blocks with rectangular sides and would not have worked in the presence of simple curved objects, rough texture-on the blocks, or without carefully controlled lighting. Nevertheless it reinforced the idea that a complete threedimensional description of the world could be extracted from a visual image. It legitimized the work of others, such as Winograd (7), whose programs worked in a make-believe world of blocks-if one program could be built which understood such a world completely and could also. manipulate that world, then it seemed that programs which assumed that abstraction could in fact be connected to the real world without great difficulty.

The role of computer-vision was "given a two-dimensional image, infer the objects that produced it, including their shapes, positions, colors, and sizes" (8). This attitude lead to an emphasis on recovery of three-dimensional shape (9), from monocular and stereo images. A number of demonstration recognition and location systems were built, such as those of Brooks (10) and Grimson (11), although they tended not to rely on using three-dimensional shape recovery.

The role of AI was to take descriptions of the world (though usually not as geometric as vision seemed destined to deliver, or as robotics seemed to need) and manipulate them based on a database of knowledge about how the world works in order to solve problems, make plans, and produce explanations. These high-level aspirations have very rarely been embodied by connection to either computer vision systems or robotics devices.

The role of robotics was to deal with the physical interactions with the world. As robotics adopted the idea of having a complete three-dimensional world model, a number of subproblems became standardized. One was to plan a collision-free path through the world model for a manipulator arm, or for a mobile robot—see the article by Yap (12) for a survey of the literature, Another was to understand forward kinematics and dynamics—given a set of joint or wheel torques as functions over time, what path would the robot hand or body follow. A more useful, but harder, problem is inverse kinematics and dynamics-given a desired trajectory as a function of time, for instance one generated by a collision-free path planning algorithm, compute the set of joint or wheel torques that should be applied to follow that path within some prescribed accuracy (13).

It became clear after a while that perfect-models of the world could not be obtained from sensors, or even CAD databases. Some attempted to model the uncertainty explicitly (14, 15) and found strategies that worked, in its presence, while others moved away from position-based techniques to force-based planning, at least in the manipulator world (16). Ambitious-plans were laid for combining many of the pieces of research over the years into a unified planning and execution system for robot manipulators (17), but after years of theoretical progress and long-term impressive engineering, the most advanced systems are stiff far from the ideal (18).

These approaches, along with those in the mobile robot domain (19, 20), shared the *sense-model-plan-act* framework, where an iteration through the cycle could often take 1-5 minutes or more (18, 19).

The New Approach

Driven by a dissatisfaction with the performance of robots in dealing with the real world, and concerned that the complexity of run-time modeling of the world was getting out of hand, a number of people somewhat independently began around 1984 rethinking the general problem of organizing intelligence. It seemed a reasonable requirement that intelligence be reactive to dynamic aspects of the environment, that a mobile robot operate on time scales similar to those of animals and humans, and that intelligence be able to generate robust behavior in the face of uncertain sensors, an unpredictable environment, and a changing world. Some of the key realizations about the organization of intelligence were as follows:

- Agre and Chapman at MIT claimed that most of what people do in their day-to-day lives is not problem-solving or planning, but rather it is routine activity in a relatively benign, but certainly dynamic, world. Furthermore the representations an agent uses of objects in the world need not rely on naming those objects with symbols that the agent possesses, but rather can be defined through interactions of die agent with the world (21, 22).

- Rosenschein and Kaelbling at SRI International (and later at Teleos Research) pointed out that an observer can legitimately talk about an agent's beliefs and goals, even though the agent need not manipulate symbolic data structures at run time. A formal symbolic specification of the agent's design can be compiled away, yielding efficient robot programs (23, 24).

- Brooks at MIT argued that in order to really test ideas of intelligence it is important to build complete agent which operate in dynamic environments using real sensors. Internal world models that are complete representations of the external environment, besides being impossible to obtain, are not at all necessary for agents to act in a competent manner. Many of the actions of an agent are quite separable—coherent intelligence can emerge from independent subcomponents interacting in the world (25-27).

All three groups produced implementations of these ideas, using as their medium of expression a network of simple computational elements, hardwired together, connecting sensors to actuators, with a small amount of state maintained over clock ticks.

Agre and Chapman demonstrated their ideas by building programs for playing video games. The first such program was called Pengi and played a concurrently running video game program, with one protagonist and many opponents which can launch dangerous projectiles (Fig. 1). There are two components to the architecture visual routine processor (VRP), which provides input to the system, and a network of standard logic gates, which can be categorized into three components: *aspect detectors, action suggestors,* and *arbiters.* The system plays the game from the same point of view as a human playing a video game, not from the point of view of the protagonist within the game. However, rather than analyze a visual bit map, the Pengi program is presented with an iconic version. The VRP implements a version of Ullman's visual routines theory *(28),* where markers from a set of six are placed on certain icons and follow them. Operators can place a marker on the *nearest opponent,* for example, and it will track that opponent even when it is no longer the nearest. The placement of these markers was the only state in the system. Projection operators let the player predict the consequences of actions, for instance, launching a projectile. The results of the VRP are analyzed by the first part of the central network and describe certain aspects of the world. In the mind of the designer, output signals designate such things as "the protagonist is moving," "a projectile from the north is about to hit the protagonist," and so on. The next part of the network takes Boolean combinations of such signals to suggest actions, and the third stage uses a fixed priority scheme (that is, it never learns) to select the next action. The use of these types of deictic representations was a key move away-from the traditional AI approach of dealing only with named individuals in the world (for instance, *opponent-27* rather than the deictic *the-opponent-which-is-closest-to-the-protagonist,* whose objective identity may change over time) and lead to very different requirements on the sort of reasoning that was necessary to perform well in the world.

Rosenschein and Kaelbling used a robot named Flakey, which operated in the regular and unaltered office areas of SRI in the vicinity of the special environment for Shakey that had been built two decades earlier. Their architecture was split into a perception subnetwork and an action subnetwork. The networks were ultimately constructed of standard logic gates and delay elements (with feedback loops these provided the network with state), although the programmer wrote at a much higher level of abstraction-in terms of goals that the robot should try to satisfy. By formally specifying the relationships between sensors and effectors and the world, and by using off-line symbolic computation, Rosenschein and Kaelbling's high-level languages were used to generate provably correct, real-time programs for Flakey. The technique may be limited by the computational complexity of the symbolic compilation process as the programs get larger and by the validity of their models of sensors and actuators.

Brooks developed the subsumption architecture, which deliberately changed the modularity from the traditional AI approach. Figure 2 shows a vertical decomposition into task achieving behaviors rather than information processing modules. This architecture was used on robots which explore, build maps, have an onboard manipulator, walk, interact with people, navigate visually, and learn to coordinate many conflicting internal behaviors. The implementation substrate consists of networks of message-passing augmented finite state machines (AFSMs). The messages are sent over predefined "wires" from a specific transmitting to a specific receiving AFSM. The messages are simple numbers (typically 8 bits) whose meaning depends on the designs of both the transmitter and the receiver. An AFSM has additional registers which hold the most recent incoming message on any particular wire. The registers can have their values fed into a local combinatorial circuit to produce new values for registers or to provide an output message. The network of AFSMs is totally asynchronous, but individual AFSMs can have fixed duration monostables which provide for dealing with the flow of time in the outside world. The behavioral competence of the system is improved by adding more behavior-specific network to the existing network. This process is called layering. This is a simplistic and crude analogy to evolutionary development. As with evolution, at every stage of the development the systems are tested. Each of the layers is a behavior-producing piece of network in its own right, although it may implicitly rely on the presence of earlier pieces of network. For instance, an *explore* layer does not need to explicitly avoid obstacles, as the designer knows that the existing *avoid* layer will take care of it. A fixed priority arbitration scheme is used to handle conflicts.

These architectures were radically different from those in use in the robotics community at the time. There was no central model of the world explicitly represented within the systems. There was no implicit separation of data and computation-they were both distributed over the same network of elements. There were no pointers, and no easy way to implement them, as there is in symbolic programs. Any search space had to be a bounded in size a priori, as search nodes could not be dynamically created and destroyed during a search process. There was no central locus of control. In general, the separation into perceptual system, central system, and actuation system was much less distinct than in previous approaches, and indeed in these systems there was an intimate intertwining of aspects of all three of these capabilities. There was no

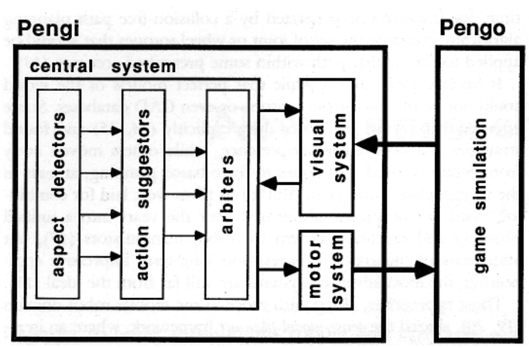

Fig. 1. The Pengi system (21) played a video game called Pengo. The control system consisted of a network of logic gates, organized into a visual system, a central system, and a motor system. The only state was within the visual system. The network within the central system was organized into three components: an aspect detector subnetwork, an action suggestor subnetwork, and an arbiter subnetwork.

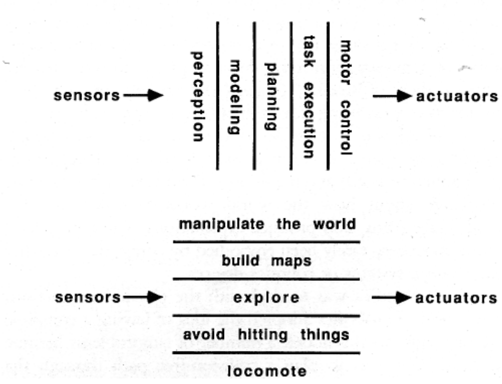

Fig. 2. The traditional decomposition for an intelligent control system within AI is to break processing into a chain of information processing modules (**top**) proceeding from sensing to action. In the new approach (**bottom**) the decomposition is in terms of behavior-generating modules each of which connects sensing to action. Layers are added incrementally, and newer layers may depend on earlier layers operating successfully, but do not call them as explicit subroutines.

notion of one process calling on another as a subroutine. Rather, the networks were designed so that results of computations would simply be available at the appropriate location when needed. The boundary between computation and the world was harder to draw as the systems relied heavily on the dynamics of their interactions with the world to produce their results. For instance, sometimes a physical action by the robot would trigger a change in the world that would be perceived and cause the next action, in contrast to directly executing the two actions in sequence.

Most of the behavior-based robotics work has been done with implemented physical robots. Some has been done purely in software (21), not as a simulation of a physical robot, but rather as a computational experiment in an entirely make-believe domain to explore certain critical aspects of the problem. This contrasts with traditional robotics where many demonstrations are performed only on software simulations of robots.

Areas of Work

Perhaps inspired by this early work and also by Minsky's (29) rather more theoretical Society of Mind ideas on how the human mind is organized, various groups around the world have pursued behavior-based approaches to robotics over the last few years. The following is a survey of some of that work and relates it to the key issues and problems for the field.

One of the shortcomings in earlier approaches to robotics and AI was that reasoning was so slow that systems that were built could not respond to a dynamic real world. A key feature of the new approaches to robotics is that the programs are built with short connections between sensors and actuators, making it plausible, in principle at least, to respond quickly to changes in the world.

The first demonstration of the subsumption architecture was on the robot Allen (25). The robot was almost entirely reactive, using sonar readings to keep away from moving *people* and other moving obstacles, while not colliding with static obstacles. It also had a non-reactive higher level layer that would select a goal to head toward, and then proceed in that direction while the lower level reactive layer took care of avoiding obstacles. It thus combines non-reactive capabilities with reactive ones. More importantly, it used exactly the same sorts of computational mechanism to do both. In looking at the network of the combined layers there was no obvious partition into lower and higher level components based on the type of information flowing on the connections, or the finite state machines that were the computational elements. To be sure, there was a difference in function between the two layers, but there was no need to introduce any centralization or explicit representations to achieve a later, higher level process having useful and effective influence over an earlier, lower level.

The subsumption architecture was generalized (30) so that some of the connections between processing elements could implement a retina bus, a cable that transmitted partially processed images from one site to another within the system. It applied simple difference operators, and region-growing techniques, to segment the visual field into moving and nonmoving parts, and into floor and nonfloor parts. Location, but not identity of the segmented regions, was used to implement image-coordinate-based navigation. All the visual techniques were known to be very unreliable on single gray-level images, but by having redundant techniques operating in parallel and rapidly switching between them, robustness was achieved. The robot was able to follow corridors and moving objects in real time, with very little computational resources by modem computer vision standards.

This idea of using redundancy over many images is in contrast to the approach in traditional computer vision research of trying to extract the maximal amount of information from a single image, or pair of images. This lead to trying to get complete depth maps over a full field of view from a single pair of stereo images. Ballard (31) points out that humans do not do this, but rather servo their two eyes to verge on a particular point and then extract relative depth information about that point. With this and many other examples he points out that an active vision system, that is, one with control over its cameras, can work naturally in object-centered coordinates, whereas a passive vision system, that is, one which has no control over its cameras, is doomed to work in viewer-centered coordinates. A large effort is under way at Rochester to exploit behavior-based or animate vision. Dickmanns and Graefe (32) in Munich have used redundancy from multiple images, and multiple feature windows that track relevant features between images, while virtually ignoring the rest of the image, to control a truck driving on a freeway at over 100 kilometers per hour.

Although predating the emphasis on behavior-based robots, Raibert's hopping robots (33) fit their spirit. Traditional walking robots are given a desired trajectory for their body and then appropriate leg motions are computed. In Raibert's one-, two-, and four-legged machines, he decomposed the problem into independently controlling the hopping height of a leg, its forward velocity, and the body attitude. The motion of the robot's body emerges from the interactions of these loops and the world. Using subsumption, Brooks programmed a six-legged robot, Genghis (Fig. 3), to walk over rough terrain (34). In this case, layers of behaviors implemented first the ability to stand up, then to walk without feedback then to adjust for rough terrain and obstacles by means of force feedback, then to modulate for this accommodation based on pitch and roll inclinometers. The trajectory for the body is not specified explicitly, nor is there any hierarchical control. The robot successfully navigates rough terrain with very little computation. Figure 4 shows the wiring diagram of the 57 augmented finite state machines that controlled it.

There have been a number of behavior-based experiments with robot manipulators. Connell (35) used a collection of 17 AFSMs to control an arm with two degrees of freedom mounted on a mobile base. When parked in front of a soda can, whether at floor level or on a table top, the arm was able to reliably find it and pick it up, despite other clutter in front of and under the can, using its local sensors to direct its search.

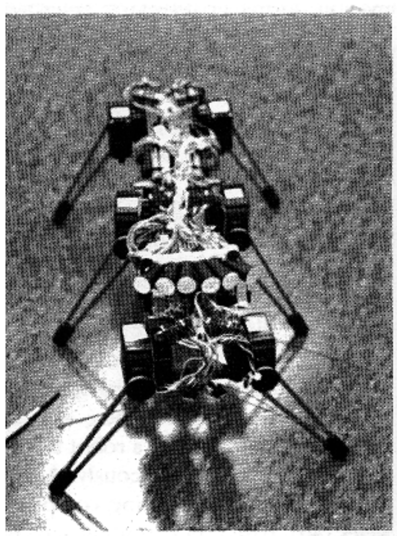

Fig. 3. Genghis is a six-legged robot measuring 35 centimeters in length. Each rigid leg is attached at a shoulder joint with two degrees of rotational freedom, each driven by a model airplane position controllable servo motor. The sensors are pitch and roll inclinometers, two collision-sensitive antennae, six forward-looking passive pyroelectric infrared sensors, and crude force measurements from the servo loops of each motor. There are four onboard eight-bit microprocessors, three of which handle motor and sensor signals and one of which runs the subsumption architecture.

All the AFSMs had sensor values as their only inputs and, as output, actuator, commands that then went through a fixed priority arbitration network to control the arm and hand. In this case, there was no communication between the AFSMs, and the system was completely reactive to its environment. Malcolm and Smithers (36) at Edinburgh report a hybrid assembly system. A traditional AI planner produces plans for a robot manipulator to assemble the components of some artifact, and a behavior-based system executes the plan steps. The key idea is to give the higher level planner robust primitives which can do more than carry out simple motions, thus making the planning problem easier.

Representation is a cornerstone topic in traditional AI. Mataric at MIT has recently introduced active representations into the subsumption architecture (37). Identical subnetworks of AFSMs are the representational units. In experiments with a sonar-based officeenvironment navigating robot named Toto, landmarks were broadcast to the representational substrate as they were encountered. A previously unallocated subnetwork would become the representation for that landmark and then take care of noting topological neighborhood relationships, setting up expectation as the robot moved through previously encountered space, spreading activation energy for path planning to multiple goals, and directing the robot's motion during goal-seeking behavior when in the vicinity of the landmark. In this approach the representations and the ways in which they are used are inseparable—it all happens in the same computational units within the network.. Nehmzow and Smithers (38) at Edinburgh have also experimented with including representations of landmarks, but their robots operated in a simpler world of plywood enclosures. They used self-organizing networks to represent knowledge of the world, and appropriate influence on the current action of the robot. Additionally, the Edinburgh group has done a number of experiments with reactivity of robots, and with group dynamics among robots using a Lego-based rapid prototyping system that they have developed.

Many of the early behavior-based approaches used a fixed priority scheme to decide which behavior could control a particular actuator at which time. At Hughes, an alternative voting scheme was produced (39) to enable a robot to take advantage of the outputs of many behaviors simultaneously. At Brussels a scheme for selectively activating and de-activating complete behaviors was developed by Maes (40), based on spreading activation within the network itself. This scheme was further developed at MIT and used to program Toto amongst other robots. In particular, it was used to provide a learning mechanism on the six-legged robot Genghis, so that it could learn to coordinate its leg lifting behaviors, based on negative feedback from falling down (41).

Very recently there has been work at IBM (42) and Teleos Research (43) using Q-1earning (44) to modify the behavior of robots. There seem to be drawbacks with the convergence time for these algorithms, but more experimentation on real systems is needed.

A number of researchers from traditional robotics (45) and AI (46, 47) have adopted the philosophies of the behavior-based approaches as the bottom of two-level systems as- shown in Fig. 5. The idea is to let a reactive behavior-based system take care of the real time issues involved with interacting with the world while a more traditional AI system sits on top, making longer term executive decisions that affect the policies executed by the lower level. Others (48) argue that purely behavior-based systems are all that are needed.

Evaluation

It has been difficult to evaluate work done under the banner of the new approaches to robotics. Its proponents have often argued on the basis of performance of systems built within its style. But performance is hard to evaluate, and there has been much criticism that the approach is both unprincipled and will not scale well. The unprincipled argument comes from comparisons to traditional academic robotics, and the scaling argument comes from traditional AI. Both these disciplines have established but informal criteria for what makes a good and respectable piece of research.

Traditional academic robotics has worked in a somewhat perfect domain. There are CAD-like models of objects. and robots, and a modeled physics of how things interact (16). Much of the work is in developing algorithms that guarantee certain classes of results in the modeled world. Verifications are occasionally done with real robots (18) but typically those trials are nowhere nearly as complicated as the examples that can be handled in simulation. The sticking point seems to be in how well the experimenters are able to coax the physical robots to match the physics of the simulated robots.

For the new approaches to robotics, however, where the emphasis is on understanding and exploiting the dynamics of interactions with the world, it makes sense to measure and analyze the systems as they are situated in the world. In the same way modern ethology has prospered by studying animals in their native habitats, not just in Skinner boxes. For instance, a particular sensor, under ideal experimental conditions, may have a particular resolution. Suppose the sensor is a sonar. then to measure its resolution an experiment will be set up where a return signal from the test article is sensed, and the resolution will be compared against measurements of distance made with a ruler or some such device. The experiment might be done for a number of different surface types. But when that sensor is installed on a mobile robot, situated in a cluttered, dynamically changing world, the return signals that reach the sensor may come from many possible sources. The object nearest the sensor may not be made of one of the tested materials. It may be at such an angle that the sonar pulse acts as though it were a mirror, and so the sonar sees a secondary reflection. The secondary lobes of the sonar might detect something in a cluttered situation where there was no such interference in the clean experimental situation. One of the main points of the new approaches to robotics is that these effects are extremely important on the overall behavior of a robot. They are also extremely difficult to model. So the traditional robotics approach of proving correctness in an abstract model may be somewhat meaningless in the new approaches. We need to find ways of formalizing our understanding the dynamics of interactions with the world so that we can build theoretical tools that will let us make predictions about the performance of our new robots.

In traditional AI there are many classes of research contributions (as distinct from application deployment). Two of the most popular are described here. One is to provide a formalism that is consistent for some level of description of some aspect of the world, for example, qualitative physics, stereotyped interactions between speakers, or categorizations or taxonomies of animals. This class of work does not necessarily require any particular results, theorems, or working programs to be judged adequate; the formalism is the important contribution. A second class of research takes some input representation of some aspects of a situation in the world and makes a prediction. For example, it might be in the form of a plan to effect some change in the world, in the form of the drawing of an analogy with some schema in a library in order to deduce some non-obvious fact, or it might be in the form of providing some expert-level advice. These research contributions do not have to be tested in situated systems—there is an implicit understanding among researchers about what is reasonable to "tell" the systems in the input data.

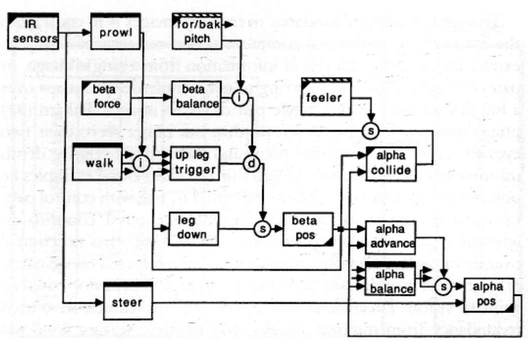

Fig. 4. The subsumption network to control Genghis consists of 57 augmented finite state machines, with "wires" connecting them that pass small integers as messages. The elements without bands on top are repeated six times, once for each leg. The network was built incrementally starting in the lower right comer, and new layers were added, roughly toward the upper left comer, increasing the behavioral repertoire at each stage.

In the new approaches there is a much stronger feeling that the robots must find everything out about their particular world by themselves. This is not to say that a priori knowledge cannot be incorporated into a robot, but that it must be non-specific to the particular location in which the robot will be tested. Given the current capabilities of computer perception, this forces behavior-based robots to operate in much more uncertain and much more coarsely described worlds than traditional AI systems operating in simulated, imagined worlds. The new systems can therefore seem to have much more limited abilities. I would argue (48), however, that the traditional systems operate in a way that will never be transportable to the real worlds that the situated behavior-based robots already inhabit.

The new approaches to robotics have garnered a lot of interest, and many. people are starting to work on their various aspects. Some are trying to build systems using only the new approaches, others are trying to integrate them with existing work, and of course there is much work continuing in the traditional style. The community is divided on the appropriate approach, and more work needs to be done in making comparisons in order to understand the issues better.

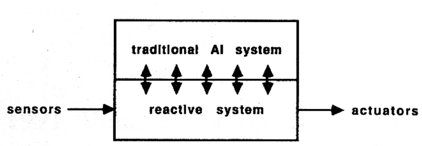

Fig. 5. A number of projects involve combining a reactive system, linking sensors, and actuators with a traditional AI system that does symbolic reasoning in order to tune the parameters of the situated component.

REFERENCES AND NOTES

1. P. H. Winston, *Artificial* Intelligence (Addison-Wesley, Reading, MA, ed. 2, 1984).

2. A. M. Turing, in *Machine* Intelligence, B. Meltzer and D. Michie, Eds. (American Elsevier, New York, 1970), vol. 5, pp. 3-23. This paper was originally written in 1948 but was not previously published.

3. Applied robotics for industrial automation has not been so closely related to Artificial Intelligence.

4. P. R. Cohen, *AI Magazine 12,* 16 (1991).

5. N. J. Nilsson, Ed., *Technical Note No. 323* (SRI International, Menlo Park, CA, 1984). This is a collection of papers and technical notes, some previously unpublished, from the late 1960s and early 1970s.

6. P. H. Winston, in *Machine* Intelligence, B. Meltzer and D. Michie, Eds. (Wiley, New York, 1972), vol. 7, pp. 431-463.

7. T. Winograd, *Understanding Natural Language* (Academic Press, New York, 1972).

8. E. Charniak and D. McDermott, *Introduction to Artificial* Intelligence (Addison Wesley, Reading, MA, 1984).

9. D. Marr, Vision (Freeman, San Francisco, 1982).

10. R. A. Brooks, *Model-Based Computer* Vision (UMI Re-search Press, Ann Arbor, 1984).

11. W. E. L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints* (MIT Press, Cambridge, MA, 1990).

12. C. K. Yap, in *Advances in Robotics,* J. T. Schwartz and C. K. Yap, Eds. (Lawrence Earlbaum, 1985), vol. 1.

13. M. Brady, J. Hollerbach, T. Johnson, T. Lozano-Pérez, M. Mason, Eds., *Robot Motion: Planning and Control* (MIT Press, Cambridge, MA, 1982).

14. R. A. Brooks, *Int. J. Robotics Res. 1,* 29 (1982).

15. R. Chatila and J.-P. Laumond, in *Proceedings of the IEEE Conference on Robotics and Automation,* St. Louis (IEEE Press, New York, 1985), pp. 138-143.

16. T. Lozano-Pérez, M. T. Mason, R. H. Taylor, *Int. J. Robotics Res.* 3, 3 (1984).

17. T. Lozano-Pérez and R. A. Brooks, in *Solid Modeling by Computers,* M. S. Pickett and J. W. Boyse, Eds. (Plenum, New York, 1984), pp. 293-327.

18. T. Lozano-Pérez, J. L. Jones, E. Mazer, P. A. O'Donnell, *Computer 22,* 21 (1989).

19. H. P. Moravec, *Proc. IEEE* 71, 872 (1982).

20. R. Simmons and E. Krotkov, in Proceedings of the *IEEE* Robotics and Automation, Sacramento (IEEE Press, New York, 1991), pp. 2086-2091.

21. P. Agre and D. Chapman, in Proceedings of American Association of Artificial Intelligence, Seattle (Morgan Kaufmann, Los Altos, CA, 1990), pp. 268-272.

22. in Designing Autonomous Agents, P. Maes, Ed. (MIT Press, Cambridge, MA, 1990), pp. 17-34.

23. S. J. Rosenschein and L. P. Kaelbling, in Proceedings of the Conference on Theoretical Aspects of Reasoning about Knowledge, J. Halpern, Ed. (Morgan Kaufmann, Los Altos, CA, 1986), pp. 83-98.

24. L. P. Kaelbling and S. J. Rosenschein, in Designing Autonomous Agents, P. Maes, Ed. (MIT Press, Cambridge, MA, 1990), pp. 35-48.

25. R. A, Brooks, *IEEE J.* Robotics Automation 2, 14 (1986).

26. —, in Designing Autonomous Agents, P. Maes, Ed. (MIT Press, Cambridge, MA, 1990), pp. 3-15.

27. —, Artificial Intelligence 47, 139 (1991).

28. S. Ullman, Cognition 18, 97 (1984).

29. M. Minsky, The Society of Mind (Simon and Schuster, New York, 1986).

30. I. D. Horswill and R. A. Brooks, in Proceedings of American Association of Artificial Intelligence, St. Paul (Morgan Kaufmann, Los Altos, CA, 1988), pp. 796-800.

31. D. H. Ballard, in Proceedings of the International Joint Conference on Artificial Intelligence, Detroit (Morgan Kaufmann, Los Altos, CA, 1989), pp. 1635-1641.

32. E. D. Dickmanns and V. Graefe, Machine Vision Appl. 1, 223 (1988).

33. M. H. Raibert, Legged Robots that Balance (MIT Press, Cambridge, MA, 1986).

34. R. A. Brooks, Neural Computation 1, 253 (1989).

35. J. H. Connell, Technical Report No. AIM-TR-1151 (MIT, Cambridge, MA, 1989).

36. C. Malcolm and T. Smithers, in Designing Autonomous Agents, P. Maes, Ed. (MIT Press, Cambridge, MA, 1990), pp. 123-144.

37. M. J. Mataric, Technical Report No. AIM-TR-1228 (MIT, Cambridge, MA, 1990).

38. U. Nehmzow and T. Smithers, in From Animals to Animats, J.-A. Meyer and S. W. Wilson, Eds. (MIT Press, Cambridge, MA, 1990), pp. 152-159.

39. D. W. Payton, in Proceedings of the *IEEE* Conference on Robotics and Automation, San Francisco (Morgan Kauffman, Los Altos, CA, 1986), pp. 1838-1843.

40. P. Maes, in Proceedings of the International Joint Conference on Artificial Intelligence, Detroit (Morgan Kauffman, Los Altos, CA, 1989), pp. 991-997.

41. P. Maes and R. A. Brooks, in Proceedings of the American Association of Artificial Intelligence, Boston (Morgan Kauffman, Los Altos, CA, 1990), pp. 796-802.

42. S. Mahadevan and J. H. Connell, Automatic Programming of Behavior-based Robots using Reinforcement Learning (IBM T. J. Watson Research Center, 1990).

43. L. Kaelbling, thesis, Stanford University (1990).

44. C. Watkins, thesis, Cambridge University (1989).

45. R. C. Arkin, in Designing Autonomous Agents, P. Maes, Ed. (MIT Press, Cambridge, MA, 1990), pp. 105-122.

46. T. M. Mitchell, in Proceedings of the American Association of Artificial Intelligence, Boston (Morgan Kauffman, Los Altos, CA, 1990), pp. 1051-1058.

47. P. K. Malkin and S. Addanki, in ibid., pp. 1045-1050.

48. R. A. Brooks, in Proceedings of the International Joint Conference on Artificial Intelligence, Sydney (Morgan Kauffman, Los Altos, CA, 1990), pp. 569-595.

49. Supported in part by the University Research Initiative under Office of Naval Research contract N00014-86-K-0685, in part by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-K-0124, in part by the Hughes Artificial Intelligence Center, in part by Siemens Corporation, and in part by Mazda Corporation.