15
Intelligence without Representation

Rodney A. Brooks
1991

1 Introduction

Artificial intelligence started as a field whose goal was to replicate human-level intelligence in a machine. Early hopes diminished as the magnitude and difficulty of that goal was appreciated. Slow progress was made over the next 25 years in demonstrating isolated aspects of intelligence. Some recent work has tended to concentrate on commercializable aspects of"intelligent assistants" for human workers.

No one talks about replicating the full gamut of human intelligence anymore. Instead we see a retreat into specialized subproblems, such as knowledge representation, natural language understanding, vision, or even more specialized areas such as truth maintenance or plan verification. All the work in these subareas is benchmarked against the sorts of tasks humans do within those areas. Amongst the dreamers still in the field of AI (those not dreaming about dollars, that is) there is a feeling that one day all these pieces will fall into place and we will see "truly" intelligent systems emerge.

However, I and others believe that human-level intelligence is too complex and too little understood to be correctly decomposed into the right subpieces at the moment, and that even if we knew the subpieces we still wouldn't know the right interfaces between them. Furthermore we will never understand how to decompose human-level intelligence until we've had a lot of practice with simpler intelligences.

In this paper I therefore argue for a different approach to creating artificial intelligence.

- We must incrementally build up the capabilities of intelligent systems, having *complete* systems at each step, thus automatically ensuring that the pieces and their interfaces are valid.

- At each step, we should build complete intelligent systems that we let loose in the real world with real sensing and real action.

> Anything less provides a candidate with which we can delude ourselves.

We have been following this approach and have built a series of autonomous mobile robots. We have reached an unexpected conclusion (**C**) and have a rather radical hypothesis (**H**).

> (**C**) When we examine very simple level intelligence we find that explicit representations and models of the world simply get in the way. It turns out to be better to let the world itself serve as its own model.
>
> (**H**) Representation is the wrong unit of abstraction in building the bulkiest parts of intelligent systems.

Representation has been the central issue in artificial intelligence work over the last 15 years only because it has provided an interface between otherwise isolated modules and conference papers.

2 The evolution of intelligence

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 scant 19,000 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 pretty simple once the essence of acting 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.

I believe 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. Moravec (1984) argues this same case rather eloquently.

Human level intelligence has provided us with an existence proof, but we must be careful about what lessons are to be gained from it.

*A **story***

> Suppose it is the 1890's. Artificial flight is the glamor subject in science, engineering, and venture capital circles. A bunch of AF researchers are miraculously transported by a time machine to the 1990's for a few hours. They spend the whole time in the passenger cabin of a commercial passenger Boeing 747 on a medium duration flight.
>
> Returned to the 1890's they feel invigorated, knowing that AF is possible on a grand scale. They immediately set to work duplicating what they have seen. They make great progress in designing pitched seats, double pane windows, and know that if only they can figure out those weird 'plastics' they will have the grail within their grasp. (A few connectionists amongst them caught a glimpse of an engine with its cover off and they are preoccupied with inspirations from that experience.)

3 Abstraction as a dangerous weapon

Artificial intelligence researchers are fond of pointing out that AI is often denied its rightful successes. The popular story goes that when nobody has any good idea of how to solve a particular sort of problem (for example, playing chess) it is known as an AI problem. When an algorithm developed by AI researchers successfully tackles such a problem, however, AI detractors claim that since the problem was solvable by an algorithm, it wasn't really an AI problem after all. Thus AI never has any successes.

> But have you ever heard of an AI failure?

I claim that AI researchers are guilty of the same (self-)deception. They partition the problems they work on into two components. The AI component, which they solve, and the non-AI component which they don't solve. Typically, AI "succeeds" by defining the parts of the

problem that are unsolved as not AI. The principal mechanism for this partitioning is abstraction. Its application is usually considered part of good science, and not (as it is in fact used in AI) as a mechanism for self-delusion. In AI, abstraction is usually used to factor out all aspects of perception and motor skills. I argue below that these are the hard problems solved by intelligent systems, and further that the shape of solutions to these problems constrains greatly the correct solutions of the small pieces of intelligence which remain.

Early work in AI concentrated on games, geometrical problems, symbolic algebra, theorem proving, and other formal systems (see the classic papers in Feigenbaum and Feldman 1963 and Minsky 1968). In each case, the semantics of the domains were fairly simple.

In the late sixties and early seventies, the "blocks world" became a popular domain for AI research. It had a uniform and simple semantics. The key to success was to represent the state of the world completely and explicitly. Search techniques could then be used for planning within this well-understood world. Learning could also be done within the blocks world; there were only a few simple concepts worth learning, and they could be captured by enumerating the set of subexpressions which must be contained in any formal description of a world containing an instance of the concept. The blocks world was even used for vision research and mobile robotics, as it provided strong constraints on the perceptual processing necessary (see, for instance, Nilsson 1984).

Eventually, criticism surfaced that the blocks world was a "toy world" and that within it there were simple special purpose solutions to what should be considered more general problems. At the same time there was a funding crisis within AI (both in the US and the UK, the two most active places for AI research at the time). AI researchers found themselves forced to become relevant. They moved into more complex domains, such as trip planning, going to a restaurant, medical diagnosis, and such like.

Soon there was a new slogan: "Good representation is the key to AI" (as in: *conceptually efficient programs,* Bobrow and Brown 1975). The idea was that by representing only the *pertinent* facts explicitly, the semantics of a world (which on the surface was quite complex) were reduced to a simple closed system once again. Abstraction to only the relevant details thus simplified the problems.

Consider chairs, for example. While these two characterizations are true,

(CAN (SIT-ON PERSON CHAIR)), and

(CAN (STAND-ON PERSON CHAIR)),

there is really much more to the concept of a chair. Chairs have some flat (maybe) sitting place, with perhaps a back support. They have a range of possible sizes, requirements on strength, and a range of possibilities in shape. They often have some sort of covering material—unless they are made of wood, metal or plastic. They sometimes are soft in particular places. They can come from a range of possible styles. In sum, the concept of what is a chair is hard to characterize simply. There is certainly no AI vision program that can find arbitrary chairs in arbitrary images; they can at best find one particular type of chair in carefully selected images.

This characterization, however, is perhaps the correct AI representation for solving certain problems—for instance, one in which a hungry person sitting on a chair in a room can see a banana hanging from the ceiling just out of reach. Such problems are never posed to AI systems by showing them a photo of the scene. A person (even a young child) can make the right interpretation of the photo and suggest a plan of action. For AI planning systems, however, the experimenter is required to abstract away most of the details to form a simple description in terms of atomic concepts such as PERSON, CHAIR and BANANA.

But this abstraction process is the essence of intelligence and the hard part of the problems being solved. Under the current scheme, the abstraction is done by the researchers, leaving little for the AI programs to do but search. A truly intelligent program would study the photograph, perform the abstraction itself, and solve the problem.

The only input to most AI programs is a restricted set of simple assertions deduced from the real data by humans. The problems of recognition, spatial understanding, dealing with sensor noise, partial models, and the like, are all ignored. These problems are relegated to the realm of input black boxes. Psychophysical evidence suggests they are all intimately tied up with the representation of the world used by an intelligent system.

There is no clean division between perception (abstraction) and reasoning in the real world. The brittleness of current AI systems attests to this fact. For example, MYCIN (Shortliffe 1976) is an expert at diagnosing human bacterial infections; but it really has no model of what a human (or any living creature) is or how they work, or what are

plausible things to happen to a human. If told that the aorta is ruptured and the patient is losing blood at the rate of a pint every minute, MYCIN will try to find a bacterial cause of the problem.

Thus, because we still perform all the abstractions for our programs, most AI work is still done in the equivalent of the blocks world. Now the blocks are slightly different shapes and colors, but their underlying semantics have not changed greatly.

It could be argued that performing this abstraction (perception) for AI programs is merely the normal reductionist use of abstraction common in all good science. The abstraction reduces the input data so that the program experiences the same "perceptual world" (what von Uexküll 1921 called a *Merkwelt)* as humans. Other (vision) researchers will independently fill in the details at some other time and place. I object to this on two grounds. First, as von Uexkiill and others have pointed out, each animal species, and clearly each robot species with its own distinctly nonhuman sensor suites, will have its own different *Merkwelt.* Second, the *Merkwelt* we humans provide our programs is based on our own introspection. It is by no means clear that such a *Merkwelt* is anything like what we actually use internally—it could just as easily be an output coding for communication purposes (thus, most humans go through life never realizing they have a large blind spot almost in the center of their visual fields).

The first objection warns of the danger that reasoning strategies developed for the human-assumed *Merkwelt* may not be valid when real sensors and perceptual processing are used. The second objection says that, even with human sensors and perception, the *Merkwelt* may not be anything like that used by humans. In fact, it may be the case that our introspective descriptions of our internal representations are completely misleading and quite different from what we really use.

A continuing story

> Meanwhile our friends in the 1890's are busy at work on their AF machine. They have come to agree that the project is too big to be worked on as a single entity and that they will need to become specialists in different areas. After all, they had asked questions of fellow passengers on their flight and discovered that the Boeing Co. employed over 6000 people to build such an airplane.
>
> Everyone is busy, but there is not a lot of communication between the groups. The people making the passenger seats used the finest solid steel available as the framework. There was some

muttering that perhaps they should use tubular steel to save weight, but the general consensus was that if such an obviously big and heavy airplane could fly then clearly there was no problem with weight.

On their observation flight, none of the original group managed a glimpse of the driver's seat, but they have done some hard thinking and believe they have established the major constraints on what should be there and how it should work. The pilot, as he will be called, sits in a seat above a glass floor so that he can see the ground below so he will know where to land. There are some side mirrors so he can watch behind for other approaching airplanes. His controls consist of a foot pedal to control speed (just as in these new fangled automobiles that are starting to appear), and a steering wheel to turn left and right. In addition the wheel stem can be pushed forward and back to make the airplane go up and down. A clever arrangement of pipes measures airspeed of the airplane and displays it on a dial. What more could one want? Oh yes. There's a rather nice setup of louvers in the windows so that the driver can get fresh air without getting the full blast of the wind in his face.

An interesting sidelight is that all the researchers have by now abandoned the study of aerodynamics. Some of them had intensely questioned their fellow passengers on this subject and not one of the modern flyers had known a thing about it. Clearly the AF researchers had previously been wasting their time in its pursuit.

4 Incremental intelligence

I wish to build completely autonomous mobile agents that co-exist in the world with humans, and are seen by those humans as intelligent beings in their own right. I will call such agents *Creatures.* This is my intellectual motivation. I have no particular interest in demonstrating how human beings work—although humans, like other animals, are interesting objects of study in this endeavor, inasmuch as they are successful autonomous agents. I have no particular interest in applications; it seems clear to me that, if my goals can be met, then the range of applications for such Creatures will be limited only by our (or their) imagination. I have no particular interest in the philosophical implications of Creatures, although clearly there will be significant implications.

Given the caveats of the previous two sections, and considering the parable of the AF researchers, I am convinced that I must tread carefully in this endeavor to avoid some nasty pitfalls.

For the moment then, consider the problem of building Creatures as an engineering problem. We will develop an *engineering methodology* for building Creatures.

First, let us consider some of the requirements for our Creatures.

- A Creature must cope appropriately and in a timely fashion with changes in its dynamic environment.

- A Creature should be robust with respect to its environment. Minor changes in the properties of the world should not lead to total collapse of the Creature's behavior; rather one should expect only a gradual change in capabilities of the Creature as the environment changes more and more.

- A Creature should be able to maintain multiple goals and, depending on the circumstances it finds itself in, change which particular goals it is actively pursuing; thus it can both adapt to surroundings and capitalize on fortuitous circumstances.

- A Creature should do *something* in the world; it should have some purpose in being.

Now, let us consider some of the valid engineering approaches to achieving these requirements. As in all engineering endeavors, it is necessary to decompose a complex system into parts, build the parts, and then interface them into a complete system.

4.1 Decomposition by function

Perhaps the strongest traditional notion of intelligent systems (at least implicitly among AI workers) has been of a central system, with perceptual modules as inputs and action modules as outputs. The perceptual modules deliver a symbolic description of the world and the action modules take a symbolic description of desired actions and make sure they happen in the world. The central system then is a symbolic information processor.

Traditionally, work in perception (and vision is the most commonly studied form of perception) and work in central systems has been done by different researchers and even totally different research laboratories. Vision workers are not immune to earlier criticisms of AI workers. Most vision research is presented as a transformation from one image representation (such as a raw grey-scale image) to another registered image (such as an edge image). Each group, AI and vision, makes assumptions about the shape of the symbolic interfaces. Hardly anyone has ever connected a vision system to an intelligent central system.

Thus the assumptions independent researchers make are not forced to be realistic. There is a real danger from pressures to neatly circumscribe the particular piece of research being done.

The central system must also be decomposed into smaller pieces. We see subfields of artificial intelligence such as "knowledge representation", "learning", "planning", "qualitative reasoning", etc. The interfaces between these modules are also subject to intellectual abuse.

When researchers working on a particular module get to choose both the inputs and the outputs that specify the module requirements, I believe there is little chance the work they do will fit into a complete intelligent system.

This bug in the functional decomposition approach is hard to fix. One needs a long chain of modules to connect perception to action. In order to test any of them, they all must first be built. But until realistic modules are built, it is highly unlikely that we can predict exactly what modules will be needed or what interfaces they will need.

4.2 Decomposition by activity

An alternative decomposition makes no distinction between peripheral systems, such as vision, and central systems. Rather, the fundamental slicing up of an intelligent system is in the orthogonal direction, dividing it into *activity* producing subsystems. Each activity, or behavior-producing system, individually connects sensing to action. We refer to an activity producing system as a *layer.* An activity is a pattern of interactions with the world. Another name for our activities might well be *skills*—since each activity can, at least post facto, be rationalized as pursuing some purpose. We have chosen the word 'activity', however, because our layers must decide when to act for themselves—not be some subroutine to be invoked at the beck and call of some other layer. We call Creatures that are decomposable into activities or behavior-producing layers in this way *behavior-based systems.*

The advantage of this approach is that it gives an incremental path from very simple systems to complex autonomous intelligent systems. At each step of the way, it is only necessary to build one small piece, and interface it to an existing, working, complete intelligence.

The idea is to build first a very simple complete autonomous system, and *test it in the real world.* Our favorite example of such a system is a Creature, actually a mobile robot, which avoids hitting things. It senses objects in its immediate vicinity and moves away from them, halting if it senses something in its path. It is still necessary to build

this system by decomposing it into parts, but there need be no clear distinction between a "perception system", a "central system" and an "action system". In fact, there may well be two independent channels connecting sensing to action—one for initiating motion, and one for emergency halts—so there is no single place where "perception" delivers a representation of the world in the traditional sense.

Next we build an incremental layer of intelligence which operates in parallel to the first system. It is pasted onto the existing debugged system and tested again in the real world. This new layer might directly access the sensors and run a different algorithm on the delivered data. The first-level autonomous system continues to run in parallel, and unaware of the existence of the second level. For example, in Brooks (1986) we reported on building a first layer of control which let the Creature avoid objects, and then adding a layer which instilled an activity of trying to visit distant visible places. The second layer injected commands to the motor control part of the first layer, directing the robot towards the goal; but, independently, the first layer would cause the robot to veer away from previously unseen obstacles. The second layer monitored the progress of the Creature and sent updated motor commands, thus achieving its goal without being explicitly aware of obstacles, which had been handled by the lower level of control.

5 Who has the representations?

With multiple layers, the notion of perception delivering a description of the world gets blurred even more, as the part of the system doing perception is spread out over many pieces which are not particularly connected by data paths or related by function. Certainly there is no identifiable place where the "output" of perception can be found. Furthermore, totally different sorts of processing of the sensor data proceed independently and in parallel, each affecting the overall system activity through quite different channels of control.

In fact, not by design but rather by observation, we note that a common theme in the ways in which our layered and distributed approach helps our Creatures meet our goals is that there is no central representation.

- Low-level simple activities can instill the Creature with reactions to dangerous or important changes in its environment. Without complex representations and the need to maintain those representations and

reason about them, these reactions can easily be made quick enough to serve their purpose. The key idea is to sense the environment often, and so have an up-to-date idea of what is happening in the world.

- By having multiple parallel activities, and by removing the idea of a central representation, there is less chance that any given change in the class of properties enjoyed by the world can cause total collapse of the system. Rather, one might expect that a given change will at most incapacitate some but not all of the levels of control. Gradually, as a more alien world is entered (alien in the sense that the properties it holds are different from the properties of the world in which the individual layers were debugged) the performance of the Creature might continue to degrade. By not trying to have an analogous model of the world, centrally located in the system, we are less likely to have built in a dependence on that model being completely accurate. Rather, individual layers extract only those *aspects* (Agre and Chapman 1987) of the world which they find relevant—projections of a representation into a simple subspace, if you like. Changes in the fundamental structure of the world have less chance of being reflected in every one of those projections than they would have of showing up as a difficulty in matching some query to a single central world model.

- Each layer of control can be thought of as having its own implicit purpose (or goal, if you insist). Since they are *active* layers, running in parallel and with access to sensors, they can monitor the environment and decide on the appropriateness of their goals. Sometimes goals can be abandoned when circumstances seem unpromising, and other times fortuitous circumstances can be taken advantage of. The key idea here is to use *the world itself as its own best model,* and to match the preconditions of each goal continuously against the real world. Because there is separate hardware for each layer, we can match as many goals as can exist in parallel; we do not pay any price for higher numbers of goals, as we would if we tried to add more and more sophistication to a single processor, or even some multi-processor with a capacity-bounded network.

- The purpose of the Creature is implicit in its higher level purposes, goals, or layers. There need be no explicit representation of goals that some central (or distributed) process selects from, to decide what is most appropriate for the Creature to do next.

5.1 No representation versus no central representation

Just as there is no central representation, there is not even a central system. Each activity-producing layer connects perception to action

directly. It is only the observer of the Creature who imputes a central representation or central control. The Creature itself has none; it is a collection of competing behaviors. Out of the local chaos of their interactions, there emerges, in the eye of an observer, a coherent pattern of behavior. There's no central, purposeful locus of control. (Minsky 1986 gives a similar account of how human behavior is generated.)

Note carefully that we are not claiming that chaos is a necessary ingredient of intelligent behavior. Indeed, we advocate careful engineering of all the interactions within the system (evolution had the luxury of incredibly long time scales and enormous numbers of individual experiments, and thus perhaps was able to do without this careful engineering).

We do claim, however, that there need be no explicit representation of either the world or the intentions of the system to generate intelligent behaviors for a Creature. Without such explicit representations, and when viewed locally, the interactions may indeed seem chaotic and without purpose.

I claim there is more than this, however. Even at a local level, we do not have traditional AI representations. We never use tokens which have any semantics that can be attached to them. The best that can be said in our implementations is that a number is passed from one process to another. But it is only by looking at the state of both the first and second processes that that number can be given any interpretation at all. An extremist might say that we really do have representations, but they are just implicit. With an appropriate mapping of the complete system and its state to another domain, we could define representations that these numbers and topological connections between processes somehow encode.

However we are not happy with calling such things representations. They differ from standard representations in too many ways.

There are no variables that need instantiation in reasoning processes. (See Agre and Chapman 1987 for a more thorough treatment of this.) There are no rules that need to be selected through pattern matching. There are no choices to be made. To a large extent, the state of the world determines the action of the Creature. Simon (1969/81) noted that the complexity of behavior of a system was not necessarily inherent in the complexity of the Creature, but perhaps in the complexity of the environment. He made this analysis in his description of an ant wandering the beach, but ignored its implications in the next paragraph when he talked about humans. We hypothesize (following

Agre and Chapman) that much of even human-level activity is similarly a reflection of the world through very simple mechanisms without detailed representations.

6 The methodology in practice

In order to build systems based on an activity decomposition so that they are truly robust, we must rigorously follow a careful methodology.

6.1 Methodological maxims

First, it is vitally important to test the Creatures we build *in the real world*—the same world that we humans inhabit. It is disastrous to fall into the temptation of testing them in a simplified world first, even with the best intentions of later transferring activity to an unsimplified world. With a simplified world (matte painted walls, rectangular vertices everywhere, colored blocks as the only obstacles) it is very easy to build a submodule of the system that happens accidentally to rely on some of those simplified properties. This reliance can then easily be reflected in the requirements on the interfaces between that submodule and others. The disease spreads and the complete system depends in a subtle way on the simplified world. When it comes time to move to the unsimplified world, we gradually and painfully realize that every piece of the system must be rebuilt. Worse than that, we may need to rethink the total design, as the issues may change completely. We are not so concerned that it might be dangerous to test simplified Creatures first, and later add more sophisticated layers of control, because evolution has been successful using this approach.

Second, as *each* layer is built, it must be tested *extensively* in the real world. The system must interact with the real world over extended periods. Its behavior must be observed and be carefully and thoroughly debugged. When a second layer is added to an existing layer, there are three potential sources of bugs: the first layer, the second layer, and the interaction of the two layers. Eliminating the first of these sources of bugs as a possibility makes finding bugs much easier. Furthermore, there remains only one thing that it is possible to vary in order to fix the bugs—the second layer.

6.2 An instantiation of the methodology: Allen

We have now built a series of robots based on the methodology of task decomposition. They all operate in an unconstrained dynamic world

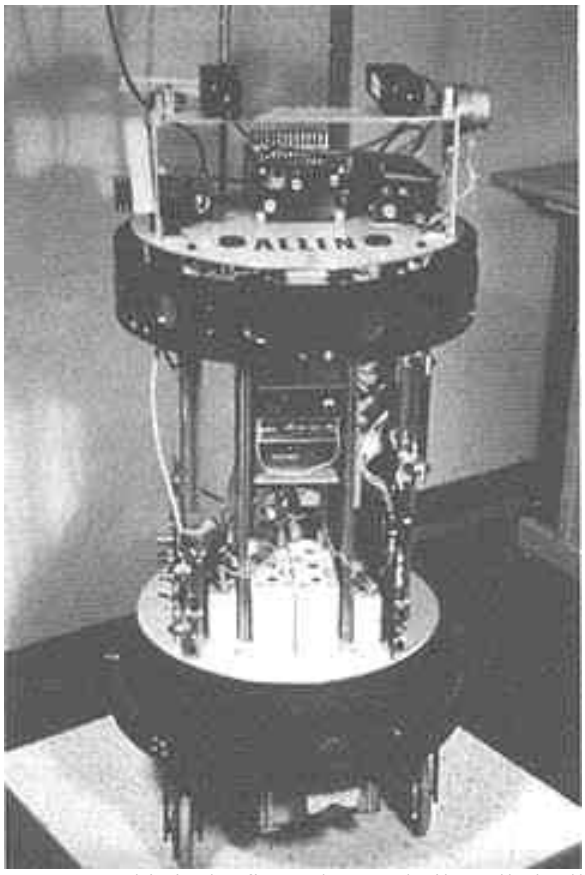

Figure 15.1: This is the first robot we built, called Allen.

(laboratory and office areas in the MIT Artificial Intelligence Laboratory). They successfully operate with people walking by, people deliberately trying to confuse them, and people just standing around watching them. All these robots are Creatures in the sense that, on power-up, they exist in the world and interact with it, pursuing multiple goals determined by their control layers implementing different activities. This is in contrast to other mobile robots that are given programs or plans to follow for a specific mission.

Our first robot, named *Allen,* is shown in figure 15.1. Allen uses an offboard Lisp machine for most of its computations. Allen implements the abstract architecture that we call the *subsumption architecture,* embodying the fundamental ideas of decomposition into layers of task-achieving behaviors, and incremental composition through debugging in the real world. (Details of this and other implementations can be found in Brooks 1987.)

Each layer in the subsumption architecture is composed of a fixed-topology network of simple finite state machines. Each finite state machine has a handful of states, one or two internal registers, one or two internal timers, and access to simple computational machines which can compute things such as vector sums. The finite state machines run asynchronously, sending and receiving fixed-length (in this case, 24-bit) messages over *wires.* For Allen, these were virtual wires; on our later robots we have used physical wires to connect computational components.

There is no central locus of control. Rather, the finite state machines are data-driven by the messages they receive. The arrival of messages or the expiration of designated time periods cause the finite state machines to change state. The finite state machines have access to the contents of the messages and might output them, test them with a predicate and conditionally branch to a different state, or pass them to simple computation elements. There is no possibility of access to global data, nor of dynamically established communications links. There is thus no possibility of global control. All finite state machines are equal, yet at the same time they are prisoners of their fixed-topology connections.

Layers are combined through mechanisms we call *suppression* (whence the name 'subsumption architecture') and *inhibition.* In both cases, as a new layer is added, one of the new wires is side-tapped into an existing wire. A predefined time constant is associated with each side-tap. In the case of suppression, the side-tapping occurs on the input side of a finite state machine. If a message arrives on the new wire, it is directed to the input port of the finite state machine as though it had arrived on the existing wire. Additionally any new messages on the existing wire are suppressed (that is, rejected) for the specified time period. For inhibition, the side-tapping occurs on the output side of a finite state machine. A message on the new wire simply inhibits messages being emitted on the existing wire for the specified time period. Unlike suppression, the new message is not delivered in their place.

As an example, consider the three layers of figure 15.2. These are three layers of control that we have run on Allen for well over a year. The robot has a ring of 12 ultrasonic sonars as its primary sensors. Every second, these sonars are run to give twelve radial depth measurements. Sonar is extremely noisy due to many objects being mirrors to sonar. There are thus problems with specular reflection and return

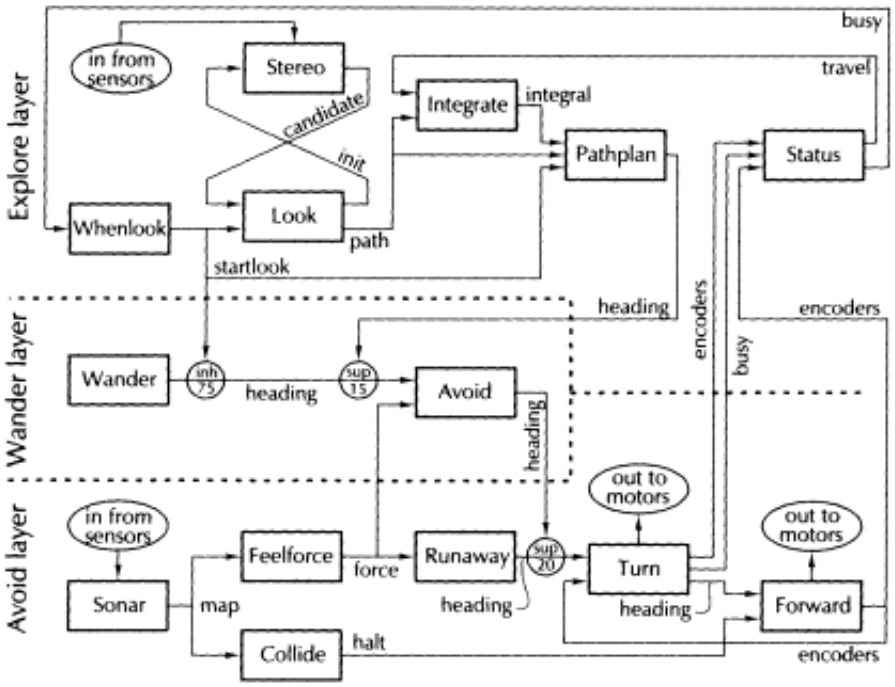

Figure 15.2: We wire finite state machines together into layers of control. Each layer is built on top of existing layers. Lower layers never rely on the existence of higher-level layers. (This is Allen.)

paths following multiple reflections due to surface skimming with low angles of incidence (less than thirty degrees).

In more detail the three layers work as follows.

1. The lowest-level layer implements a behavior which makes the robot (the physical embodiment of the Creature) avoid hitting objects. It avoids both static objects and moving objects—even those that are actively attacking it. The finite state machine labelled *sonar* simply runs the sonar devices and every second emits an instantaneous map with the readings converted to polar coordinates. This map is passed on to the *collide* and *feelforce* finite state machines. The first of these simply watches to see of there is anything dead ahead, and if so sends a *halt* message to the finite state machine in charge of running the robot forwards. (If that finite state machine is not in the correct state the message may well be ignored.) Simultaneously, the other finite state machine computes a repulsive force on the robot, based on an inverse-square law, where each sonar return is considered to indicate the presence of a repulsive object. The contributions from all the sonars are

vector-added to produce an overall force acting on the robot. The output is passed to the *runaway* machine, which thresholds it and passes it on to the *turn* machine, which orients the robot directly away from the summed repulsive force. Finally the *forward* machine drives the robot forward. Whenever this machine receives a halt message while the robot is driving forward, it commands the robot to halt.

This network of finite state machines generates behaviors which let the robot avoid objects. If it starts in the middle of an empty room it simply sits there. If someone walks up to it, the robot moves away. If it moves in the direction of other obstacles it halts. Overall, it manages to exist in a dynamic environment without hitting or being hit by objects.

2. The next layer makes the robot wander about, when not busy avoiding objects. The *wander* finite state machine generates a random heading for the robot every ten seconds or so. The *avoid* machine treats that heading as an attractive force and sums it with the repulsive force computed from the sonars. It uses the result to suppress the lower-level behavior, forcing the robot to move in a direction close to what *wander* decided but at the same time avoiding any obstacles. Note that if the *turn* and *forward* finite state machines are busy running the robot, the new impulse to wander will be ignored.

3. The third layer makes the robot try to explore. It looks for distant places, then tries to reach them. This layer suppresses the wander layer, and observes how the bottom layer diverts the robot due to obstacles (perhaps dynamic). It corrects for any divergences, and the robot achieves the goal.

The *whenlook* finite state machine notices when the robot is not busy moving, and starts up the free space finder (labelled *stereo* in the diagram) finite state machine. At the same time it inhibits wandering behavior so that the observation will remain valid. When a path is observed it is sent to the *pathplan* finite state machine, which injects a commanded direction to the *avoid* finite state machine. In this way lower-level obstacle avoidance continues to function. This may cause the robot to go in a direction different from that desired by *pathplan.* For that reason, the actual path of the robot is monitored by the *integrate* finite state machine, which sends updated estimates to the *pathplan* machine. This machine then acts as a difference engine, forcing the robot in the desired direction and compensating for the actual path of the robot as it avoids obstacles.

These are just the particular layers that were first implemented on Allen. (See Brooks 1986 for more details; Brooks and Connell 1986 report on another three layers implemented on that particular robot.)

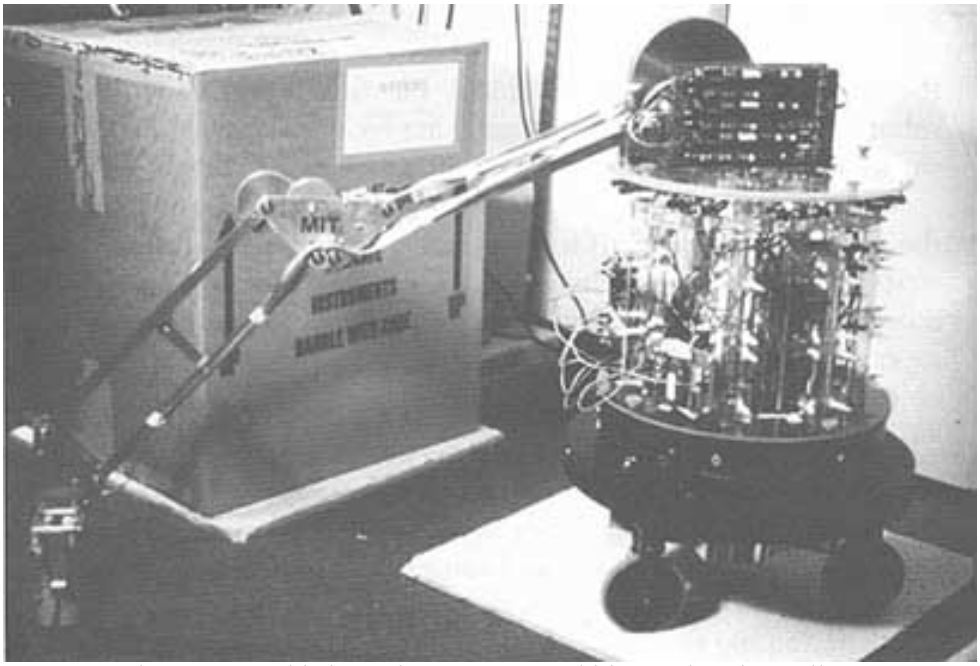

Figure 15.3: This is Herbert, a more ambitious robot than Allen.

6.3 A second example: Herbert

Allen's lowest layer was entirely reactive: it merely avoided collisions. But its next two layers, *wander* and *explore,* were not entirely reactive. Our second Creature, a mobile robot named *Herbert* (Connell 1989), was a much more ambitious project, and pushed the idea of reactivity—as in Allen's lowest layer—much further.

Herbert (shown in figure 15.3) used thirty infrared proximity sensors to navigate along walls and through doorways, a magnetic compass to maintain a global sense of direction, a laser scanner to find soda-can-like objects visually, and a host of sensors on an arm with a set of fifteen behaviors which, together, were sufficient to locate and pick up soda cans reliably. Herbert's task was to wander around people's offices looking for soda cans, pick one up, and bring it back to where the robot had started from. Herbert did succeed at this task (although mechanical failures in the seating of its onboard chips limited reliable operation to about fifteen minutes at a time).

In programming Herbert, it was decided that it should maintain no internal state longer than three seconds, and that there would be no internal communication between behavior generating modules. Each one was connected to sensors on the input side, and a fixed-priority

arbitration network on the output side. The arbitration network drove the actuators.

Since Herbert maintained hardly any internal state—hardly any memory—it often had to rely on the world itself as its only available "model" of the world. Further, the world itself was the only effective medium of communication between Herbert's separate modules. The laser-based soda-can finder, for example, 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 such that, eventually, the robot would pick up the soda can.

The advantage of this approach was that there was no need to set up internal expectations for what was going to happen next. That meant that the control system could both (1) be naturally opportunistic if fortuitous circumstances presented themselves, and (2) easily respond to changed circumstances—such as some other object approaching 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.

The point of Herbert is two-fold:

- It demonstrates complex, apparently goal-directed and intentional behavior in a system which has no long-term internal state and no internal communication; and

- It is very easy for an observer of such a system to attribute more complex internal structure than really exists—Herbert, for instance, appeared to be doing things like path planning and map building, even though it was not.

7 What this is not

The subsumption architecture with its network of simple machines is reminiscent, at the surface level at least, of a number of mechanistic approaches to intelligence, such as connectionism and neural networks. But it is different in many respects from these endeavors, and also quite different from many other post-Dartmouth* traditions in artificial intelligence. We very briefly explain those differences in the following paragraphs.

7.1 It isn't connectionism

Connectionists try to make networks of simple processors. In that regard, the things they build (in simulation only—no connectionist system has ever driven a real robot in a real environment, no matter how simple) are similar to the subsumption networks we build. However, their processing nodes tend to be uniform, and they seek insights (as their name suggests) from learning how best to interconnect them (which is usually assumed to mean richly, at least). Our nodes, by contrast, are all unique finite state machines, the density of connections among them is much lower, is not at all uniform, and is especially low between layers. Additionally, connectionists seem to be looking for explicit distributed representations to arise spontaneously from their networks. We harbor no such hopes because we believe representations are not necessary and appear only in the eye or mind of the observer.

7.2 It isn't neural networks

Neural-network research is the parent discipline, of which connectionism is a recent incarnation. Workers in neural networks claim that there is some biological significance to their network nodes, as models of neurons. Most of the models seem wildly implausible given the paucity of modeled connections relative to the thousands found in real neurons. We claim no biological significance in our choice of finite state machines as network nodes.

---

* *Editor's note:* Newell and Simon presented the first working AI program, *The Logic Theorist,* at a famous workshop organized by John McCarthy at Dartmouth College in the summer of 1956.

7.3 It isn't production rules

Each individual activity-producing layer of our architecture could be viewed as in implementation of a production rule. When the right conditions are met in the environment, a certain action will be performed. We feel that analogy is a little like saying that any FORTRAN program with IF statements is implementing a production-rule system. But a production system really is more than that—it has a rule base, from which a particular rule is selected by matching the preconditions for some or all of the rules to a given database; and these preconditions may include variables which must be bound to individuals in that database. Our layers, on the other hand, run in parallel and have no variables or need for matching. Instead, aspects of the world are extracted and directly trigger or modify certain behaviors of the layer.

7.4 It isn't a blackboard

If one really wanted, one could make an analogy of our networks to a blackboard control architecture. Some of the finite state machines would be localized knowledge sources. Others would be processes acting on these knowledge sources by finding them on the blackboard. There is a simplifying point in our architecture however: all the processes know exactly where to look on the "blackboard", since they are hardwired to the correct place. I think this forced analogy indicates its own weakness. There is no flexibility at all in where a process can gather appropriate knowledge. Most advanced blackboard architectures make heavy use of the general sharing and availability of almost all knowledge. Furthermore, in spirit at least, blackboard systems tend to hide from a consumer of knowledge who the particular producer was. This is the primary means of abstraction in blackboard systems. In our system we make such connections explicit and permanent.

7.5 It isn't German philosophy

In some circles, much credence is given to Heidegger as one who understood the dynamics of existence. Our approach has certain similarities to work inspired by this German philosopher (for instance, Agre and Chapman 1987) but our work was not so inspired. It is based purely on engineering considerations. That does not preclude it from being used in philosophical debate as an example on any side of any fence, however.

8 **Key ideas**

Situatedness, embodiment, intelligence, and emergence can be identified as key ideas that have led to the new style of artificial intelligence research that we are calling "behavior-based robots".

8.1 Situatedness

Traditional artificial intelligence has adopted a style of research where the agents that are built to test theories about intelligence are essentially problem solvers that work in a symbolic abstracted domain. The symbols may have referents in the minds of the builders of the systems, but there is nothing to ground those referents in any real world. Furthermore, the agents are not situated in a world at all. Rather, they are simply given a problem, and they solve it. Then they are given another problem, and they solve that one. They are not participating in a *world* at all, as do agents in the usual sense.

In these systems, there is no external world per se, with continuity, surprises, or history. The programs deal only with a model world, with its own built-in physics. There is a blurring between the knowledge of the agent and the world it is supposed to be operating in. Indeed, in many artificial intelligence systems, there is no distinction between the two: the agent is capable of direct and perfect perception as well as direct and perfect action. When consideration is given to porting such agents or systems to operate in the world, the question arises of what sort of representation they need of the real world. Over the years within traditional artificial intelligence, it has become accepted that they will need an objective model of the world with individuated entities, tracked and identified over time. The models of knowledge representation that have been developed expect and require such a one-to-one correspondence between the world and the agent's representation of it.

Early AI robots, such as Shakey and the Cart, certainly followed this approach. They built models of the world, planned paths around obstacles, and updated their estimates of where the objects were relative to themselves as they moved. We have developed a different approach (Brooks 1986) in which a mobile robot uses the world itself as its own model—continuously referring to its sensors rather than to an internal world model. The problems of object class and identity disappear. The perceptual processing becomes much simpler. And the performance of this robot (Allen) is better in comparable tasks than

the Cart. (The tasks carried out by Allen, not to mention Herbert, are in a different class from those attempted by Shakey—Shakey could certainly not have done what Allen does.)

A situated agent must respond in a timely fashion to its inputs. Modelling the world completely under these conditions can be computationally challenging. But a world in which it is situated also provides some continuity to the agent. That continuity can be relied upon, so that the agent can use its perception of the world instead of an objective world model. The representational primitives that are useful then change quite dramatically from those in traditional artificial intelligence.

The key idea from situatedness is: *The world is its own best model.*

8.2 Embodiment

There are two reasons that embodiment of intelligent systems is critical. First, only an embodied intelligent agent is fully validated as one that can deal with the real world. Second, only through a physical grounding can any internal symbolic or other system find a place to bottom out, and give "meaning" to the processing going on within the system.

The physical grounding of a robot within the world forces its designer to deal with all the issues. If the intelligent agent has a body, has sensors, and has actuators, then all the details and issues of being in the world must be faced. It is no longer possible to argue in conference papers that the simulated perceptual system is realistic, or that problems of uncertainty in action will not be significant. Instead, physical experiments can be done simply and repeatedly. There is no room for "cheating" (in the sense of self-delusion). When this is done, it is usual to find that many of the problems that used to seem significant are not so in the physical system. Typically, "puzzle-like" situations, where symbolic reasoning had seemed necessary, tend not to arise in embodied systems. At the same time, many issues that had seemed like nonproblems become major hurdles. Typically, these concern aspects of perception and action. (In fact, there is some room for cheating even here: for instance, the physical environment can be specially simplified for the robot—and it can be very hard in some cases to identify such self-delusions.)

Without an ongoing participation in and perception of the world, there is no meaning for an agent—everything is empty symbols referring only to other symbols. Arguments might be made that, at some

level of abstraction, even the human mind operates in this solipsist position. However, biological evidence suggests that the human mind's connection to the world is so strong, and so many-faceted, that these philosophical abstractions may not be correct.

The key idea from embodiment is: *The world grounds the regress of meaning-giving.*

8.3 Intelligence

Earlier, I argued that the sorts of activities we usually think of as demonstrating intelligence in humans have been taking place for only a very small fraction of our evolutionary lineage. I argued further that the "simple" things concerning perception and mobility in a dynamic environment took evolution much longer to perfect, and that all those capabilities are a necessary basis for "higher-level" intellect.

Therefore, I proposed looking at simpler animals as a bottom-up model for building intelligence. It is soon apparent, when "reasoning" is stripped away as the prime component of a robot's intellect, that the dynamics of the interaction of the robot and its environment are primary determinants of the structure of its intelligence.

Simon's (1969) discussion of the ant walking along a beach started off in a similar vein. He pointed out that the complexity of the behavior of the ant is more a reflection of the complexity of its environment than of its own internal complexity. He speculated that the same might be true of humans—but then, within two pages of text, reduced the study of human behavior to the domain of crypt-arithmetic problems.

It is hard to draw a line between what is intelligence and what is environmental interaction. In a sense, it doesn't really matter which is which, inasmuch as all intelligent systems must be situated in some world or other if they are to be successful or useful entities.

The key idea from intelligence is: *Intelligence is determined by the dynamics of interaction with the world.*

8.4 Emergence

In discussing where intelligence resides in an artificial intelligence program, Minsky (1961) points out that "there is never any 'heart' in a program", but rather that, if we look, "we find senseless loops and sequences of trivial operations". It is hard to point at a single component as the seat of intelligence. There is no homunculus. Rather, intelligence emerges from the interaction of the components of the system.

The way in which it emerges, however, is quite different for traditional and for behavior-based artificial intelligence systems.

In traditional artificial intelligence, the modules that are defined are information-processing or functional modules. Typically, these might include a perception module, a planner, a world modeler, a learner, and the like. Such components directly participate in the functions of perceiving, planning, modeling, learning, and so on. Intelligent behavior of the system as a whole—such as avoiding obstacles, standing up, controlling gaze, et cetera—emerges from the interaction of the components.

In behavior-based artificial intelligence, by contrast, the modules that are defined are behavior-producing. Typically, these might include modules for obstacle avoidance, standing up, gaze control, and the like. Such components directly participate in producing the behaviors of avoiding obstacles, standing up, controlling gaze, and so on. Intelligent functionality of the system as a whole—such as perception, planning, modeling, learning, et cetera—emerges from the interaction of the components.

Although this dualism between traditional and behavior-based systems looks pretty, it is not entirely accurate. Traditional systems have hardly ever been really connected to the world, and so the emergence of intelligent behavior is, in most cases, more of an expectation than an established phenomenon. Conversely, because of the many behaviors present in a behavior-based system, and their individual dynamics of interaction with the world, it is often hard to say that a particular series of actions was produced by a particular behavior-module. Sometimes many behaviors are occurring simultaneously, or are switching rapidly.

It is not feasible to identify the seat of intelligence within any system, since intelligence is produced by the interactions of many components. Intelligence can only be determined by the total behavior of the system and how that behavior appears in relation to the environment.

The key idea from emergence is: *Intelligence is in the eye of the observer.*

9 Limits to growth

Since our approach is performance based, it is the performance of the systems we build which must be used to measure its usefulness and to point to its limitations.

We claim that our behavior-based robots, using the subsumption architecture to implement complete Creatures, are by now the most reactive real-time mobile robots in existence. Most other mobile robots are still at the stage of individual "experimental runs" in static environments, or at best in completely mapped static environments. Ours, on the other hand, operate completely autonomously in complex dynamic environments at the flick of their on-switches, and continue until their batteries are drained. We believe they operate at a level closer to simple insect-level intelligence than to bacteria-level intelligence. Evolution took 3 billion years to get from single cells to insects, and only another 500 million years from there to humans. This statement is not intended as a prediction of our future performance, but rather to indicate the nontrivial nature of insect-level intelligence.

Despite this good performance to date, there are a number of serious questions about our approach. We have beliefs and hopes about how these questions will be resolved, but under our criteria only performance truly counts. Experiments and building more complex systems take time. So, in the interim, the best we can do is indicate where the main questions lie, with the hope that there is at least a plausible path forward to more intelligent machines from our current situation.

Our belief is that the sorts of activity-producing layers of control we are developing (mobility, vision, and survival related tasks) are necessary prerequisites for higher-level intelligence in the style we attribute to human beings. The most natural and serious questions concerning limits of our approach are:

- How many behavior-based layers can be built in the subsumption architecture before the interactions between layers become too complex to continue?
- How complex can the behaviors be that are developed without the aid of central representations?
- Can higher-level functions such as learning occur in these fixed topology networks of simple finite state machines?

Only experiments with real Creatures in real worlds can answer the natural doubts about our approach. Time will tell.