Intelligence without representation*

Rodney A. Brooks

MIT Artificial Intelligence Laboratory, 545 Technology Square, Rm. 836, Cambridge, MA 02139, USA

Received September 1987

Brooks, R.A., Intelligence without representation, Artificial Intelligence 47 (1991), 139–159.

* This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the research is provided in part by an IBM Faculty 9 Development Award, in part by a grant from the Systems Development Foundation, in part by the University Research Initiative under Office of Naval Research contract N00014-86-K-0685 and in part by the Advanced Research Projects Agency under Office of Naval Research contract N00014-85-K-0124.

Abstract

Artificial intelligence research has foundered on the issue of representation. When intelligence is approached in an incremental manner, with strict reliance on interfacing to the real world through perception and action, reliance on representation disappears. In this paper we outline our approach to incrementally building complete intelligent Creatures. The fundamental decomposition of the intelligent system is not into independent information processing units which must interface with each other via representations. Instead, the intelligent system is decomposed into independent and parallel activity producers which all interface directly to the world through perception and action, rather than interface to each other particularly much. The notions of central and peripheral systems evaporateeverything is both central and peripheral. Based on these principles we have built a very successful series of mobile robots which operate without supervision as Creatures in standard office environments.

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. 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 any more. Instead we see a retreat into specialized subproblems, such as ways to represent knowledge, natural language understanding, vision or even more specialized areas such as truth maintenance systems 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 all fall into place and we will see "truly" intelligent systems emerge.

However, I, and others, believe that human level intelligence is too complex and 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 level 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 of the way and thus automatically ensure 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 use the world 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 mere 10,000 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 being and reacting are available. That essence is the ability to move around in a dynamic environment, sensing the surroundings to a degree sufficient to achieve the necessary maintenance of life and reproduction. This part of intelligence is where evolution has concentrated its time—it is much harder.

I believe that mobility, acute vision and the ability to carry out survivalrelated tasks in a dynamic environment provide a necessary basis for the development of true intelligence. Moravec [11] argues this same case rather eloquently.

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

2. 1. A story

Suppose it is the 1890s. 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 1980s 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 1890s they feel vigorated, 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 their 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 (e.g. 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, 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 (e.g. [6, 9]). 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 including 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 [12].

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, etc.

Soon there was a new slogan: "Good representation is the key to AI" (e.g. *conceptually efficient programs in* [2]). 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 a chair for example. While the following two characterizations are true:

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

there is 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 particular the concept of what is a chair is hard to characterize simply. There is certainly no AI vision program which 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 of solving certain problems; e.g., a person sitting on a chair in a room is hungry and 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 BANANAS.

But this abstraction 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 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, etc. 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 [13] 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 blocks world. Now the blocks have 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 (*Merkwelt* in [15]) 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 Uexküll and others have pointed out, each animal species, and clearly each robot species with their own distinctly non-human sensor suites, will have their 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 (e.g., 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 perception processing is 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.

3.1. A continuing story

Meanwhile our friends in the 1890s 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 to get a glimpse of the driver's seat, but they have done some hard thinking and think 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 newfangled 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 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, 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 (e.g., a raw grey scale image) to another registered image (e.g., 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 connect s 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 skill, emphasizing that 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.

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 first build a very simple complete autonomous system, and *test it in the real world.* Our favourite 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 subsystem", 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 on to 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 [3] 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* [1] 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 central single 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 be using the world as its own model and to continuously match the preconditions of each goal against the real world. Because there is separate hardware for each layer we can match as many goals as can exist in parallel, and 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 multiprocessor 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 is no central purposeful locus of control. Minsky [10] 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 implementation is that one number is passed from a 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 that they are just implicit. With an appropriate mapping of the complete system and its state to another domain, we could define a representation that these numbers and topological connections between processes somehow encode.

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

There are no variables (e.g. see [1] for a more thorough treatment of this) that need instantiation in reasoning processes. There are no rules which 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 [14] 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; i.e., in 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 accidentally build a submodule of the system which happens 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, or the interaction of the two layers. Eliminating the first of these source of bugs as a possibility makes finding bugs much easier. Furthermore, there is only one thing possible to vary in order to fix the bugs—the second layer.

6.2. An instantiation of the methodology

We have built a series of four robots based on the methodology of task decomposition. They all operate in an unconstrained dynamic world (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 by watching them. All four 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,

The four robots are shown in Fig. 1. Two are identical, so there are really three, designs. One uses an offboard LISP machine for most of its computations, two use onboard combinational networks, and one uses a custom onboard parallel processor. All the robots implement the same abstract architecture, which we call the *subsumption architecture* which embodies the fundamental ideas of decomposition into layers of task achieving behaviors, and incremental composition through debugging in the real world. Details of these implementations can be found in [3].

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 messages (1-bit messages on the two small robots, and 24-bit messages on the larger ones) over wires. On our first robot 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 pre-defined 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 net 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 (i.e., 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 Fig. 2. These are three layers of control that we have run on our first mobile robot for well over a year. The robot has a ring of twelve 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 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:

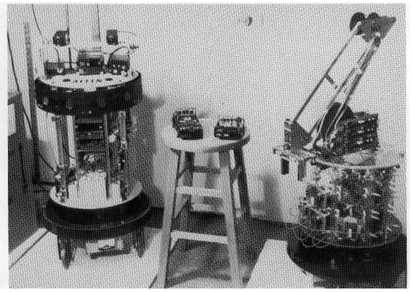

Fig. 1. The four MIT AI laboratory Mobots. Left-most is the first built Allen, which relies on an offboard LISP machine for computation support. The right-most one is Herbert, shown with a 24 node CMOS parallel processor surrounding its girth. New sensors and fast early vision processors are still to be built and installed. In the middle are Tom and Jerry, based on a commercial toy chassis, with single PALs (Programmable Array of Logic) as their controllers.

(1) The lowest-level layer implements a behavior which makes the robot (the physical embodiment of the Creature) avoid hitting objects. It both avoids 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 machine. The first of these simply watches to see if 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 each sonar are 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.

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 avoid 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.

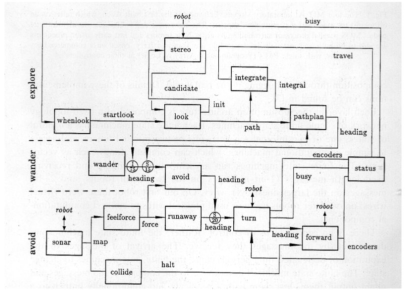

Fig. 2. We wire, finite state machines together into layers of control. Each layer is built on top of existing layers. Lower level layers never rely on the existence of higher level layers.

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 to 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 particular layers were implemented on our first robot. See [3] for more details. Brooks and Connell [5] report on another three layers implemented on that particular robot.

7. What this is not

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

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 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 are looking (as their name suggests) for revelations from understanding how to connect them correctly (which is usually assumed to mean richly at least). Our nodes are all unique finite state machines and the density of connections is very much lower, certainly not uniform, and very low indeed between layers. Additionally, connectionists seem to be looking for explicit distributed representations to spontaneously arise 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 networks 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.

7.3. It isn't production rules

Each individual activity producing layer of our architecture could be viewed as an 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. A standard production system really is more—it has a rule base, from which a rule is selected based on matching preconditions of all the rules to some database. The preconditions may include variables which must be matched to individuals in the database, but layers run in parallel and have no variables or need for matching. Instead, aspects of the world are extracted and these 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 as they are hard-wired to the correct place. I think this forced analogy indicates its own weakness. There is no flexibility at all on 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 (e.g. [1]) 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. Limits to growth

Since our approach is a performance-based one, 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 as of mid-1987 our robots, using the subsumption architecture to implement complete Creatures, are 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. Our goal (worth nothing if we don't deliver) is simple insect level intelligence within two years. 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 with the caveat that the experiments described below have not yet been performed we outline how we currently see our endeavor progressing. Our intent in discussing this is to indicate 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 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?

We outline our current thoughts on these questions.

8.1. How many layers?

The highest number of layers we have run on a physical robot is three. In simulation we have run six parallel layers. The technique of completely debugging the robot on all existing activity producing layers before designing and adding a new one seems to have been practical till now at least.

8.2. *How complex?*

We are currently working towards a complex behavior pattern on our fourth robot which will require approximately fourteen individual activity producing layers.

The robot has infrared proximity sensors for local obstacle avoidance. It has an onboard manipulator which can grasp objects at ground and table-top levels, and also determine their rough weight. The hand has depth sensors mounted on it so that homing in on a target object in order to grasp it can be controlled directly. We are currently working on a structured light laser scanner to determine rough depth maps in the forward looking direction from the robot.

The high-level behavior we are trying to instill in this Creature is to wander around the office areas of our laboratory, find open office doors, enter, retrieve empty soda cans from cluttered desks in crowded offices and return them to a central repository.

In order to achieve this overall behavior a number of simpler task achieving behaviors are necessary They include: avoiding objects, following walls, recognizing doorways and going through them, aligning on learned landmarks, heading in a homeward direction, learning homeward bearings at landmarks and following them, locating table-like objects, approaching such objects, scanning table tops for cylindrical objects of roughly the height of a soda can, serving the manipulator arm, moving the hand above sensed objects, using the hand sensor to look for objects of soda can size sticking up from a background, grasping objects if they are light enough, and depositing objects.

The individual tasks need not be coordinated by any central controller. Instead they can index off of the state of the world. For instance the grasp behavior can cause the manipulator to grasp any object of the appropriate size seen by the hand sensors. The robot will not randomly grasp just any object however, because it will only be when other layers or behaviors have noticed an object of roughly the right shape on top of a table-like object that the grasping behavior will find itself in a position where its sensing of the world tells it to react. If, from above, the object no longer looks like a soda can, the grasp reflex will not happen and other lower-level behaviors will cause the robot to look elsewhere for new candidates.

8.3. *Is learning and such possible?*

Some insects demonstrate a simple type of learning that has been dubbed "learning by instinct" [7]. It is hypothesized that honey bees for example are pre-wired to learn how to. distinguish certain classes of flowers, and to learn routes to and from a home hive and sources of nectar. Other insects, butterflies, have been shown to be able to learn to distinguish flowers, but in an information limited way [8]. If they are forced to learn about a second sort of flower, they forget what they already knew about the first, in a manner that suggests the total amount of information which they know, remains constant.

We have found a way to build fixed topology networks of our finite state machines which can perform learning, as an isolated subsystem, at levels comparable to these examples. At the moment of course we are in the very position we lambasted most AI workers for earlier in this paper. We have an isolated module of a system working, and the inputs and outputs have been left dangling.

We are working to remedy this situation, but experimental work with physical Creatures is a nontrivial and time consuming activity. We find that almost any pre-designed piece of equipment or software has so many preconceptions of how they are to be used built into them, that they are not flexible enough to be a part of our complete systems. Thus, as of mid-1987, our work in learning is held up by the need to build a new sort of video camera and high-speed low-power processing box to run specially developed vision algorithms at 10 frames per second. Each of these steps is a significant engineering endeavor which we are undertaking as fast as resources permit.

Of course, talk is cheap.

8.4. *The future*

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

Acknowledgement

Phil Agre, David Chapman, Peter Cudhea, Anita Flynn, David Kirsh and Thomas Marill made many helpful comments on earlier drafts of this paper.

References

[1] P.E. Agre and D. Chapman, Unpublished memo, MIT Artificial Intelligence Laboratory, Cambridge, MA (1986).

[2] R.J. Bobrow and J.S. Brown, Systematic understanding: synthesis, analysis, and contingent knowledge in specialized understanding systems, in: R.J. Bobrow and A.M, Collins, eds., *Representation and Understanding* (Academic Press, New York, 1975) 103-129.

[3] R.A. Brooks, A robust layered control system for a mobile robot, *IEEE J. Rob. Autom. 2* (1986) 14-23.

[41 R.A. Brooks, A hardware retargetable distributed layered architecture for mobile robot control, in: *Proceedings IEEE Robotics and Automation*, Raleigh, NC (1987) 106-110.

[5] R.A. Brooks and J.H. Connell, Asynchronous distributed control system for a mobile robot, in: *Proceedings SPIE*, Cambridge, MA (1986) 77-84.

[6] E.A. Feigenbaum and J.A. Feldman, eds., *Computers and Thought* (McGraw-Hill, San Francisco, CA, 1963).

[7] J.L. Gould and P. Marler, Learning by instinct, Sci. *Am.* (1986) 74-85.

[8] A.C. Lewis, Memory constraints and Rower choice in pieris rapae, *Science* 232 (1986) 863-865.

[9] M.L. Minsky, ed., *Semantic Information Processing* (MIT Press, Cambridge, MA, 1968).

[10] M.L. Minsky, *Society of Mind* (Simon and Schuster, New York, 1986).

[11] H.P. Moravec, Locomotion, vision and intelligence, in: M. Brady and R. Paul, eds., *Robotics Research* 1 (MIT Press, Cambridge, MA, (1984) 215-224.

[12] N.J. Nilsson, Shakey the robot, Tech. Note 323, SRI AI Center, Menlo Park, CA (1984).

[13] E.H. Shortliffe, *MYCIN: Computer-Based Medical Consultations* (Elsevier, New York, 1976).

[14] H.A. Simon, *The Sciences of the Artificial* (MIT Press, Cambridge, MA, 1969).

[15] J. Von Uexküll, *Umwelt and Innenwelt der Tiere* (Berlin, 1921).