ARTIFICIAL INTELLIGENCE-THE NEXT LEVEL

ABSTRACT

ARTIFICIAL INTELLIGENCE-THE NEXT LEVEL

The term Artificial Intelligence (AI) refers to "the science and engineering of making intelligent machines- intelligence as exhibited by an artificial (man-made, non-natural, manufactured) entity. It basically deals with the primary concept of replicating human intelligence. Representation is the current method in use in the field of AI. "Representation" or "abstraction" is the idea of representing only the pertinent facts explicitly, the semantics of a world. The human mind is capable of assembling such pertinent facts provided to it to obtain a graphic image of what is being described. In this paper, I argue that representation is not the best method to achieve AI. Because when I think of the human mind the first word that crosses my head is IMAGINATION- the innate ability to invent partial or complete personal realms within the mind from elements derived from sense perceptions of the shared world. In simple terms the extent of imagination, ideas, intelligence, depends on our interaction with the real world around us. This paper outlines the approach to build complete intelligent system that is capable of sensing and interacting with the environment around it and achieve higher levels of intelligence in the process.

SYSTEM REQUIREMENTS:

1. It should be able to sense the surroundings
2. Should be able to adapt
3. The loss of one feature or part of the system should not affect the functioning of the entire system
4. It should have some purpose

SYSTEM DEVELOPMENT:

1. Decompose the entire system into parts (independent layers).

2. Add sensors and required components to each layer.

3. Interface these layers to obtain a full working system

PAPER

ARTIFICIAL INTELLIGENCE-THE NEXT LEVEL

INTRODUCTION

Artificial intelligence started as a field whose goal was to replicate human level intelligence in a machine. Now it has become a field to create "intelligent assistants" for human workers. No one talks about replicating the full capacity of human intelligence anymore. Human level intelligence is too complex and little understood to be correctly decomposed into the right sub pieces at the moment and interfaced. 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:

1 We must incrementally build up the capabilities of intelligent systems

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

3 Hence when we examine very simple level intelligence we find that representations and models of the world simply get in the way. It turns out to be better to use the world as a model.

THE EVOLUTION OF INTELLIGENCE

We already have an existence proof of the possibility of intelligent entities, human beings. Evolution has taken place over the past 4.6 billion years.

Single cell entities----->photosynthetic plants----->fish and vertebrates----->insects----->reptiles----->dinosaurs---->mammals----->primates and apes----->man

(3.5 billion years) (2.5 billion) (1 billion) (450 million) (370 million) (330 million) (250 million) (120 million) (2.5 million)

Man then invented agriculture a mere 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, 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. I believe that mobility, vision and the ability to carry out survival related tasks in a dynamic environment provide a basis for the development of true intelligence

REPRESENTATION –KEY TO AI

It is common to say 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. The principal mechanism of AI, abstraction- is a self delusion mechanism. In AI , abstraction is usually used to factor out all aspects of perception and motor skills. 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. In each case the semantics were fairly simple. The key to success was to represent the state of the world completely and explicitly. For example when we observe an object or read a book, there are basic obvious features we remember. These features tend to represent the object, etc.

Soon there was a new slogan: "Good representation is the key to AI". 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:

A chair is an object that you can sit on. But 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.

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 for the right image after putting the pieces of input together. The only input to most AI programs is a restricted set of simple assertion deduced from the real data by humans. The problems of recognition, spatial understanding, and dealing with sensor noise, partial models, etc. are all ignored.

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 of the intelligent work is being done by us. 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 and hence you can’t really refer to the program as a truly intelligent one. We find there is no clean division between perception (abstraction) and reasoning in the real world through representation. The brittleness of current AI systems attests to this fact.

Now let us observe a different approach. Imagine yourself in the following situation. You are tied to a chair and asked to obtain a book placed on a table out of reach. Now how would you get the book? What image of the chair would you get? A wheel chair? A swivel chair? Or maybe even a remote controlled chair which you can move towards the table (while positioning yourself in the chair) to get the book.

I have come to realize that when the human mind is provided with a situation or problem to solve, thinking tends to broaden (note that our extent of imagination broadens from a simple chair to a high-fi mobile one when we input a situation rather than specific details as in the representative method) .Imagination, creativity sets in implicitly. In the above problem we find that more than the object (chair) by itself in the picture, greater importance is given to the object's surroundings (table with a book on it).Hence our thinking is not only based on an isolated object but also the world around it. A truly intelligent program would do the same- study the situation, perform the abstraction and solve the problem itself rather than we feeding it prioritized inputs. Therefore, a higher level of intelligence will be achieved.

But we can put forth a question. How is an artificially created system capable of learning?

Some insects demonstrate a simple type of learning that has been dubbed "learning by instinct". 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 information limited way. 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.

Hence we require a network of finite state machines which can perform learning, as an isolated subsystem, at levels comparable to the above example.

INCREMENTAL INTELLIGENCE

I wish to build a system that co-exists in the world with humans, and are seen by those humans as intelligent beings in their own right. If my goals can be met then the range of applications for such systems will be limited only by our (or their) imagination.

For the moment then, consider the problem of building a robot as an engineering problem. We will develop an engineering methodology for building such systems.

First, let us consider some of the requirements for our systems:

* It must cope appropriately with changes in its dynamic environment.

* It should be robust with respect to its environment; minor changes in the properties of the world should not lead to total collapse of
the system.

* It should be able to maintain multiple goals and, depending on the circumstances it finds itself in, it should be able to adapt.

* It 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 problems it is necessary to decompose
a complex system into parts, build the parts, and then interface them into a complete system.

DECOMPOSITION BY ACTIVITY

Perhaps the strongest traditional notion of intelligent systems 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 (just like a brain to a human). In such a system all the interfaced parts are connected and dependant on the central system. Failure of the central system can lead to total collapse of the entire system itself (similar to collapse of the entire human body due to brain failure). This proves to be a great disadvantage.

An alternative approach is decomposition of the entire system into parts or layers and then interfacing them. This procedure implies layer independence. Each layer will work with respect to its own makes no distinction between peripheral systems, such as vision, and central systems. Rather it is the slicing up of an intelligent system and dividing it into activity producing subsystems. Each activity or behavior producing system individually connects sensing to action. An activity producing system is a layer. An activity is a pattern of interactions with the world. The layers must decide when to act for themselves, not be some subroutine to be invoked at the call of some other layer. The advantage of this approach is that it gives an incremental path from very simple systems to complex 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. An example of such a system is 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. 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 system continues to run in parallel, and unaware of the existence of the second level. 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.

GOOD-BYE TO REPRESENTATIONS

With multiple layers, the notion of 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(as all layers are independent). 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 different channels of control.

ADVANTAGES:

* Low-level simple activities can instill the robot with reactions to 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 properties of 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 system might continue to degrade.
Rather, individual layers extract only those aspects 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

* Each layer of control can be thought of as having its own implicit purpose. 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 circumstances can be taken advantage of. The key idea here is to be using the
world as a model and to continuously match the preconditions of each goal against the real world.

* The purpose of the system 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 system to do next.

METHOD IN PRACTICE

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

First, it is important to test the system 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 (e.g. a room with colored blocks as the only obstacles) it is very easy to accidentally build a sub module 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 sub module and others. The disease spreads and the complete system depends in some 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.

In the decomposition approach we will not be so concerned that it might be dangerous to test simplified systems 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 (as each layer is independent) 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.

AN INSTANTIATION

All the robots implement the same abstract architecture, called the subsumption architecture, which embodies the fundamental ideas of decomposition into layers of task achieving behaviors, and incremental intelligence. 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. 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 causes 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 and conditionally branch to a different state, or pass them to simple computation elements. There is no possibility of access to global data, or 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.

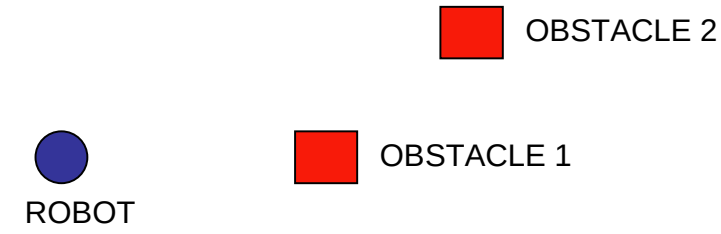
Consider the idea of a robot that comes to a stop when it encounters an object and moves to the nearest empty space (the two being separate activities independent of each other).

As you know the robot is composed of two different layers. The first layer uses an off board LISP machine for most of its computations, the second uses onboard combinational networks and a custom onboard parallel processor. CMOS parallel processors, Pals (Programmable Array of Logic) form their controller’s composition through debugging in the real world.

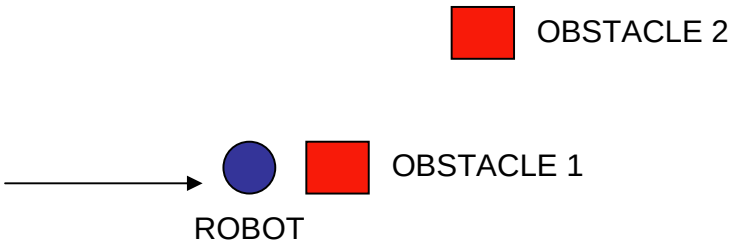
In more detail the two layers work as follows:

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. A finite state sonar device 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 feel force 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.

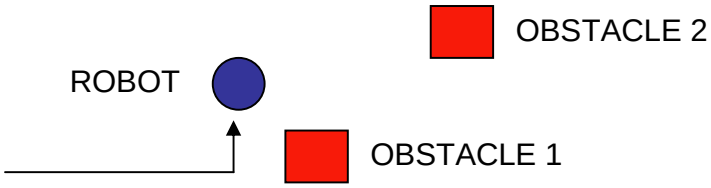
INITIAL POSITIONS



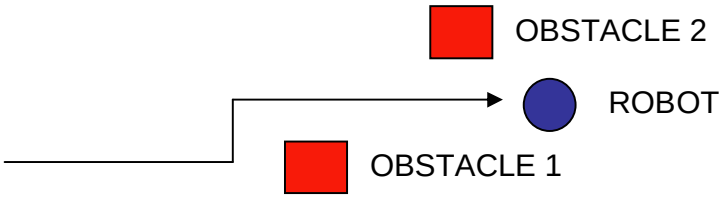
LAYER 1 ACTIVATED



LAYER 2 ACTIVATED



LAYERS 1 AND 2 ACTIVATED



LIMITATIONS

Despite the prospect of good performance, there are a number of serious questions about our approach. My 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 are 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?

THE FUTURE

Only experiments with real Creatures in real worlds can answer the natural doubts about our approach. But I'm sure it's not impossible looking back at the developments over the past few decades. All we can say is, time will tell.

If one says "four legs, a seat, back rest, you sit on it"(abstracted input), our mind would simulate the image of a simple chair (simulated output). Such images are seen with the "mind's eye".

Reference: