Artificial Autonomous Agents and Learning Systems

Bart de Boer
Free University of Brussels (Belgium)
Leiden University (the Netherlands)

November 8, 1994

Preface

These are the texts of two lectures delivered on the summer school for advanced information processing in Varna, Bulgaria, which was held from the 26th until the 30th of august 1994. The summer school was part of a bigger Tempus-exchange programme of the european community in which Bulgarian universities and universities of the Netherlands, Greece, Belgium and the United Kingdom took part.

I was a representative of the artificial intelligence laboratory of the free university of Brussels at the time. I was working there on a project involving learning systems on artificial autonomous agents. The Tempus exchange would not have been possible without the aid of Dr. Walter van de Velde (free university of Brussels) Dr. Peter Braspenning (university of Limburg, the Netherlands) and Dr. Vassil Vassilev (new Bulgarian university).

1 Part 1: Artificial Autonomous Agents

The subject of this lecture is what I consider to be one of the most challenging topics in science today: the construction of artificial autonomous agents.

This term may sound as something from a very specialistic or even a bit obscure part of science, but as we will see artificial autonomous agents could have far reaching implications in science, as well as in ethics, philosophy, biology et cetera. In fact they are related to so called artificial life, the attempt to construct the equals of living things, animals or even humans. But before making too presumptuous claims, let me define what I mean when I say: "Artificial autonomous agent".

2 Definition

An artificial autonomous agent is a man made construct that interacts with its environment in a purposeful way and on its own initiative.

First of all the thing has to be made and designed by man. This excludes slipper animacules, ants, frogs, snakes, cows et cetera. These could be considered autonomous agents, but are by no means artificial.

Secondly, the thing has to interact with its environment in a purposeful way. It both has to react to input from its environment and to perform actions in the environment that are based on its input and (possibly) on an internal plan. This excludes almost all constructs man has made until now. Many objects exist that act on their environment and that might even react to this environment. Examples of these are cars, refrigerators, televisions and even scissors and ballpoint pens, but the interactions of these have no purpose of their own. Of course one could have a lengthy discussion of what purpose is in the context of a robot, but I will trust the audience's intuition and say that a system has a purpose if it reacts to its environment in a way that can not be predicted by the directly visible interactions alone.

Finally it has to do this without (human) supervision. A lot of very interesting remotely controlled devices have been built, some of them outstanding technical achievements, but these cannot be called autonomous. Good examples of such devices are deep space crafts and the volcano exploring robot Dante[1].

Of course this definition leaves ample room for controversy. Should the organisms that result from genetic engineering be included? I would say no, as they are (as yet) only minor modifications of an existing, albeit natural, design. But should computer constructs, sometimes called software agents, that are unleashed in a computer or a computer network to perform a certain task, be included? They usually do not interact with the real world, but they definitely interact with their own virtual world in a purposeful way and they operate autonomously and without supervision.

The focus of the lecture will be artificial autonomous agents that operate in the real world and whose behaviour is determined by a computer. But the quest for creating artificial autonomous agents is much older than the computer. Let us indulge in a small and rather unscientific historical overview.

[1] David Wettergreen, Chuck Thorpe, Red Whittaker, *Exploring Mount Erebus by walking robot*, in Robotics and Autonomous Systems Vol 11, Nos. 3-4, December 1993

3 History [2]

Throughout history, man has sought to recreate life. Either by making machines or by making stories. Mythology is filled with examples of men trying to copy life. The ancient Greek had Pandorra, who had been created by a sculptor, although she could only be brought to life by the gods. In the tales of the Arabian nights an iron horse appears that can not only do what an ordinary horse can do, but that can also fly. Mediæval Prague was haunted by the Golem, a creature that had been made by Rabbi Löw. More modern examples are Frankenstein's monster and the robots of Isaac Asimov.

What most of these examples have in common is that the creation of life is seen as a challenge, but that one has to be extremely careful in doing so, or else one will cause great problems.

These stories tell us more about man's attitude towards artificial life than about his technical achievements. But there have been a lot of attempts to construct working systems that showed autonomous behaviour. The ancient Greeks already had doors that opened automatically for a visitor. Later, in the 18th century, the frenchman Vaucanson created an artificial duck, and the family Jacquet-Droz created machines that could write and make drawings. But Vaucanson's as well as Jacquet-Droz' machines did not really interact with their environment, they executed a very complicated, but unchanging program. Moreover these kinds of machines were usually curiosities and fairground-attractions and sometimes they were downright forgery.

The first machines that can really be considered artificial autonomous agents in the sense I will use in this lecture, were much more modest. They were two electric turtles, Elmer and Elsie, that were constructed by Grey Walter in the 1950's. These could react to each other and find a charging station when their batteries ran down.

From then on the construction of artificial autonomous agents was taken to hand in a more serious and a more realistic way. The direct aim was now not anymore to make a duplicate of man, but to construct machines that can operate independently in a complex environment. It appears that the creation of objects that are animated is closer at hand than ever. If done with discretion, this can become the fulfillment of an age-old dream.

4 Applications

Why would we want to create autonomous machines in the first place? There are several reasons for this. The first of these is of course scientific curiosity. Man is continuously trying to stretch the limits of what is possible and creating autonomous machines is exactly something which is at those limits. Also from the viewpoint of biology and psychology the creation of artificial autonomous agents is an interesting enterprise. By solving the problems that arise in creating such an agent, we hope to learn more about the ways nature has solved these problems when "creating" life.

But there are more practical reasons to pursue the building of artificial autonomous agents. There are many applications in which they would be extremely useful. One could think of highly dangerous jobs that nowadays can only be done by risking the health or lives of men. These jobs include the cleaning of nuclear plants, the extinguishing of burning oil wells and inspection of off-shore oil drilling platforms.

[2] Based in part on: Chris Langton, *Artificial Life* pp. 1-47 Addison Wesley, Redwood City, CA, 1989

There are also jobs that may not be dangerous, but that are impossible for man to perform. An example of this is the cleaning of sewers (a project on which the AI-lab of the free university of Brussels in cooperation with GMD in Bonn is currently working). Deep space crafts, too, could benefit from a certain degree of autonomy. Work is currently being done on an unmanned, autonomous Mars vehicle.

But one can also imagine tedious, but complex jobs being done by autonomous robots. Present robots are not very flexible in the work they do. Once they are programmed for a certain job, they can only do that. If the situation changes only a little bit, the robot will not be able to function properly anymore. Autonomy could help in such a situation. But one could also envision robots in a more domestic situation. What about a vacuum cleaning robot?

5 Techniques

Let us now proceed to the more practical side of artificial autonomous agents. There are mainly three aspects to autonomous agent design that I want to discuss. First of all, an autonomous agent needs to have sensors with which it can observe its environment. It also needs to have actuators with which it can manipulate or move through the environment. And it needs a system that transforms the inputs of the sensors into more or less reasonable outputs to the actuators. In this lecture I want to concentrate on this control system, but first I want to say something about the sensors and the actuators, because a lot of research into autonomous agents is directed towards sensors and actuators.

Sensors can be of many types. They can be optical or infra-red vision systems, they can be infra red or acoustic range finders, they can be touch sensors or speed sensors and they can even be navigational instruments that measure the position or the direction of a robot. Usually a robot is equipped with sets of sensors that complement each other, so that the strong points of one type of sensor compensate for the weak points of the other.

Most of the research efforts have gone into sensors that produce images[3]. As the images have to be processed extremely quickly, these are usually low resolution images. But although spectacular results have been reached, they belong more to the area of digital image processing than that of autonomous agents research.

The actuators usually are either meant for locomotion, i.e. the propelling of the robot through its environment, or for manipulation of this environment. Locomotion can be either by wheels or legs[4] for land-based vehicles and by jets or propellors for air or water based vehicles. All kinds of locomotion have their own advantages and disadvantages. This is for example the reason why so much research is being done to walking vehicles. Whereas wheeled vehicles are easier to build and can move faster, legged vehicles can negotiate more complex terrains.

Manipulation is done by a host of different grippers, robot arms or even snake-like flexible arms[5]. All these manipulators are beautiful examples of combinations of state-of-the-art mechanical and electronic components, usually involving sensors that sense the position of the manipulator as well. Special purpose robots can be equipped with more specialized instruments.

[3] See for example: Berthold Klaus Paul Horn, *Robot Vision*, MIT Press, Cambridge MS, 1986

[4] See for example: Song and Waldron, *Machines That Walk*, MIT Press, Cambridge, MS, 1989

[5] Shigeo Hirose, *Biologically inspired robots, snake-like locomotors and manipulators*, Oxford University Press, Oxford, 1993

There are two approaches to the construction of artificial autonomous agents. The first approach is to construct a large, multifunctional robot that has a lot of very complicated sensors and that has accurate multi-purpose manipulators. Such a robot would be usable in many different situations. These robots could be called man-like. Most of the artificial autonomous agents that have been realized are of this type.

The other approach is to construct very simple, single-purpose robots that have simple sensors and simple manipulators and that are tailored for one task only, but that can reach complexity by cooperating. These robots could be considered ant-like. This is the approach that is favoured in the AI-lab in Brussels.

Now we have mentioned some of the more mechanical aspects of artificial autonomous agents, it is time to proceed to their core: the system that determines the behaviour of the robot, which I will call the control system. The control system is usually not so much separated from the sensors and the actuators as it might appear from the way I have presented things. Most of the time the control system is adapted to the sensors and actuators it has to control. Moreover, a lot of task specific preprocessing of raw sensor data is usually conducted.

In the control system, too, there are two approaches. The first approach is the approach from classical artificial intelligence. This I will also call the sequential approach. The second approach is more of a dynamic systems approach. It could also be called a parallel approach.

The classical approach to artificial autonomous agent control is to take the inputs from the sensors, to process these, to build an internal representation, or a model of the world according to this data, then to plan an action in this model and finally to execute the planned actions.

The dynamics approach is very different from this. In this approach multiple control systems are active simultaneously. These control systems all implement a very simple part of the behaviour of the robot, but by cooperating they can build more complex behaviours. The most basic control systems implement reflexes: extremely simple reactions from actuators to certain sensory inputs or to certain internal states. Higher level control systems coordinate the reflexes, determine which one is active in which situation and which reflexes should have priority over which others. The higher level control systems can also override the outputs of the lower level ones and substitute the output of these with that of their own.

Note that this distinction appears to be similar to the distinction made between complex and simple robots. It is different, however. A simple robot can be controlled by a classical control system and a complex robot can be controlled by a dynamic control system. The similarity is only in the underlying philosophy: the achievement of complexity through cooperating simple systems. Also note that the distinction is not a black and white one. Dynamic systems will sometimes be constructed of basic control systems that work in a classical way internally and classical systems will usually have a lot of subsystems working in parallel, for example the ones controlling the movements of the actuators.

The dynamics approach is the one that is used at the AI-lab of the Free University of Brussels. Several arguments can be given in favour of the dynamics approach as opposed to the classical approach.

The control of a robot working in a real environment should be real-time. This means that the response time to changes in the sensor data must be very short. In the classical approach a long path from the sensors to the actuators exists. Furthermore, for even the smallest change in the sensors, the whole trajectory has to be followed. In the dynamics approach on the other hand, there usually exists a short path from sensors to actuators. This reflex does not have to give the best possible response, but in most cases a suboptimal response is better than a response that is too late. In more complex cases, where there is more time, the higher level control systems in the dynamic system can provide a better response.

Also a control system that is built according to the dynamics approach is easier to maintain and extend. An extension to a sequential system usually involves modifying substantial parts of the code. In the dynamics approach one only has to add some extra reflexes or an extra layer of control systems.

A last, and more subjective advantage of the dynamics approach is that it is a more biological approach. Behaviours of animals and men seem also to be organized in a dynamics way with reflexes, default actions and exceptions to the default actions. Seen from an engineer's point of view, one could say that this does not really matter, but when one is interested in the study of artificial life one prefers to investigate the more lifelike system. Besides, it is likely that natural evolution has found an effective system to solve the control problem of autonomous agents, so it is worthwhile to look into that first, before trying to invent something new.

For the dynamics approach to be useful, a programming tool is needed to write control systems in an easy way. I will now present two of these. The first one is based on the subsumption architecture, develloped by Brooks [6] of the Massachusetts Institute of Technology. The second one is called PDL, or Process Description Language and is used at the AI-lab of the Free University of Brussels where it was develloped by Luc Steels[7].

In the subsumption architecture, basic control systems are implemented by augmented finite state machines that run in parallel. Augmented finite state machines are finite state machines with a set of registers. They are augmented so that they get Turing-machine equivalent computational strength. The augmented finite state machines take input data, process this according to their internal state and create output data.

There are links between basic control systems, sensors and actuators along which data can be moved. Basic control systems can have multiple inputs, but one input can also be connected to the outputs of multiple control systems. If this is the case and both of the control system connected to the input are active an arbitration scheme is necessary. Only one of the control system is allowed to provide its data to the input. In case of conflict the arbitration always chooses the same control system. It is determined by the designer which one of the control system is allowed to provide the data.

Thus a kind of hierarchy, or subsumption, among the control systems is designed into the system. Certain control systems always have priority over others and can provide for exceptions to the more general rules that the lower priority control systems implement.

For the implementation special tools, for example the *behavior language* have been developed. The language of choice is LISP.

In PDL, the basic control systems, also called processes are implemented as small pieces of halting C or C++ code that do not have local variables. All interaction between processes, and between processes and sensors and processes and actuators is handled by so called quantities. Quantities are global variables that can be read, but not directly written by the processes. Processes can only propose to add a value to a quantity.

---

[6] Rodney A. Brooks, *A robust layered control system for a mobile robot*, IEEE Journal of Robotics and Automation, RA-2(1), April 1986

[7] Luc Steels, *The artificial life roots of artificial intelligence*, Artificial Life Journal, Vol 1,1. MIT Press, Cambridge, MS, 1994

The processes are run in parallel. All processes are started simultaneously after which they are allowed to run and finish. For all processes an account is kept of which values they want to add to which quantities. After all processes have finished all the proposed additions are added to the quantity. If a quantity exceeds a maximum or minimum value, it is set to this maximum respectively minimum value. Note that during a cycle no changes are made to a quantity.

After the processes have been ran and the quantities have been updated, the values of quantities associated with actuators are copied to these. If a quantity is associated with a sensor, the value of that sensor is copied into the quantity. The old value of the quantity is lost.

The philosophy behind PDL is to create a system in which processes have a dynamical influence, which arises in cooperation (or competition) with other processes. Furthermore all processes are equal in PDL. This in opposition to the subsumption architecture in which it is fixed beforehand which process will have priority over which other processes. Under the subsumption architecture a process will not know if its contribution is really used or not. In PDL a process can be sure that its contribution is always used.

Another difference between the two architectures is that the subsumption architecture is asynchronous, whereas PDL is synchronous. In PDL every process has ran as many times as any other process in the system; all processes are always started at the same time and a new cycle is only initiated when all processes have finished. Under the subsumption architecture processes run independently and timing problems are resolved with external timers.

Both approaches are meant to be portable among different platforms. It would be very inconvenient if software written for one robot would have to be modified extensively to be ran on a different robot. The subsumption architecture as well as PDL provide for a hardware-independent interface for the robot. It is now possible to test the same control system on completely different robots by simply recompiling the code.

The subsumption architecture as well as PDL have been used to implement behaviours of varying complexity successfully. The availability of these (and other) programming tools highly facilitates the construction of artificial autonomous agents. But much work still needs to be done to perfect the tools.

6 Learning

Another aspect of autonomous behaviour is the possibility to adapt to a changing environment and to learn new behaviours. As I will say some things about learning systems on artificial autonomous agents in the second part of this paper, I will be brief here.

Learning and adaptation in autonomous agents is essential in all but the most trivial situations. Take for example the case of a camera coordinating the movements of a robot arm. If there is no adaptation at all in this system, a small distortion of the camera or the arm will cause the system to malfunction.

But learning systems in autonomous agents have to fulfill special criteria. First of all they have to be able to learn on-line and in real time. This means that they should be able to learn while they are performing and that the learning algorithm should be so fast as not to disturb the performance of the system and to achieve results in a short time.

Furthermore the system should not overtrain itself. Some learning systems, when trained too long, will cease to function properly. They become inflexible or unstable. This should not happen in the learning system of an autonomous agent, as this is always operating. After all, one never knows when something should be learnt.

Also the learning system should be immune to noise. The sensors of a robot can not always be trusted, an environment does not always have to be deterministic and sometimes strange situations occur out of pure coincidence. A learning system should not be fooled by these things.

A last and perhaps rather obvious criterium is that the learning system should be able to learn in an unsupervised way. If one would always need to have a teacher to train the system, then one could not call it a truly autonomous system.

There exist very few learning systems that fulfill these criteria. Classical learning systems are often too sensitive to noise. Dynamic learning systems, such as neural networks[8] and classifier systems[9] are immune to noise, but usually suffer from overtraining. Genetic algorithms[10] are a promising approach, but are not readily applicable in an on-line setting.

So although learning systems are quite necessary on artificial autonomous agents, a good and safe algorithm still has to be found. Personally I find learning systems for robots one of the most interesting areas of computer science research.

7 Future

We have seen some of the techniques used to create artificial autonomous agents and we have discussed some of the problems encountered. To this date no one has succeeded in constructing a completely autonomous agent, so obviously they have also not been used yet in commercial real world applications. However, I believe that in ten or fifteen years artificial autonomous agents will start to appear in everyday life. Autonomous agents that reside in a virtual information world, or *software agents*, will probably appear even sooner. In fact software that has autonomy to initiate bank transfers already exist, for example in electronic cash dispensers.

But before this can happen a lot of problems still need to be solved. We have to develop reliable sensors and actuators, but most of all we have to develop a reliable control system. This control system should allow the robot to perform its task properly, without doing harm to itself or to its environment, to take care that it has enough resources (energy, materials) needed to do its job and to do it in such a way that human intervention is rarely necessary. Today such a control system does not yet exist.

The emergence of autonomy in robots also involves a lot of interesting legal, ethical and philosophical issues. The ethical and philosophical issues come into play especially when we start to make artifacts that are not only autonomous, but that also display intelligence. Are we allowed to create such artifacts? And if we do, for what purposes can we use them? Is it ethical to use them for military purposes? And if we create artifacts with higher levels of intelligence, shouldn't we grant these some kind of civil rights? Is it not dangerous to create machines that have a will of their own? And what will happen if we make machines that

[8] See for example: J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the Theory of neural Computation*, Addison Wesley, Reading MS, 1991

[9] J. H. Holland, *Escaping Brittleness, The possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems*, in Ryszard S. Michalski, Jaime G. Carbonell, Tom M. Michell (eds.) Machine Learning, an Artificial Intelligence Approach part II, pp. 593–623

[10] See for example: D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, Reading MS, 1989

are so successful that they threaten to supplant ourselves[11]? These questions have been discussed many times, but I think some very interesting points of view can be found in the robot stories of science fiction writers like Stanisław Lem and Isaac Asimov.

Perhaps some of the ethical and philosophical issues sound a bit like something for the far future. One could argue that we still have to go a long way before we can create machines with such levels of intelligence that we have to worry about the ethical side. Personally I think that it is never to early to worry about ethics. However, even for the most simple systems we should give thought to the legal sides[12], because otherwise we have the risk there will be no law about the technology.

If we do create and use systems with a certain autonomy we have to solve some interesting legal problems. We should determine who is responsible for autonomous systems. Will the user or the designer be responsible? Or will we ultimately have to lay the responsibility on the system itself? Because if something goes wrong, we will definitely want to lay the responsibility somewhere. Furthermore we have to assess the risks that are involved in creating and marketing a certain autonomous system. Will the benefits be bigger than the risks? If not, should we not forbid the marketing of such a system?

Room should be created in the law to grant certain powers to autonomous systems. As I have mentioned before there already exist systems that can perform financial transactions. These systems thus must have a certain power, i.e. the power to perform a financial transaction, that is usually only granted to people. The capacity of making decisions of autonomous systems will only grow with time, so their legal abilities will have to grow as well.

The creation of artificial life has always been a dream of mankind. Throughout his history man has tried to create the equal or even the superior of himself. Nowadays we almost have the technical capabilities to make this dream come true. But a lot of problems still have to be solved. These problems are not only of a technical nature. The creation of artificial autonomous agents brings about a lot of ethical and legal problems as well. And it could well be that these will eventually be found to be the biggest problems. But if we pay attention to these difficulties and avoid the many possible pitfalls, the construction of a truly autonomous and intelligent agent will be one of the biggest technical achievements of mankind.

[11] Hans Moravec, *Human Culture: A Genetic Takeover Underway*, in: Chris Langton, ed. *Artificial Life*, Addison-Wesley, Redwood City, CA, 1989

[12] Based on: John Cas Czelen, *Legal Implications of Intelligent Autonomous Systems*, in: L.O. Herzberger and F.C.A. Groen, eds. *Intelligent Autonomous Systems*, North Holland, Amsterdam, 1987

1 Part 2: Learning Systems

Ladies and gentlemen. Would it not be wonderful if we did not have to program our computers anymore? If we could just explain them what we want them to do or if we could show them some examples and they would then distill from that what we want. Or if our computers would adapt to our preferences and to our idiosyncrasies? For these things to come true, computers need, among other things of course, an ability to learn.

The ability to adapt to their environment has always been a characteristic of living things. Animals do this all the time, but also plants and lower life forms, like single-celled organisms usually adapt to their environment to a certain degree. Computer systems have traditionally been notorious for their rigidity and their inability to adapt, but this could be changed by using learning computer systems.

Of course not all applications will benefit from learning. But one could think of many situations in which computers could benefit from the ability to adapt. Especially in complex environments that are difficult to model, in real world control applications and in user interfacing there are great opportunities for applying learning systems.

2 Definition

Now that I have indicated that there is a use for learning computer systems, and before I proceed to describe their workings, I will try to clarify a bit what learning systems are.

Everybody has an intuitive understanding of what learning means. We all have learned at school and at the university, and we have acquired all kinds of other skills during our lives. In fact, man is learning throughout his live. But there are some difficulties in defining and studying learning.

Philosophers, as well as psychologists, biologists and ethologists have studied learning in animals and in humans for a long time. The phenomena they have researched not only include the acquisition of new skill and knowledge, but also much simpler things, like habituation (getting used to a foreign stimulus) and adaptation (modification of already present behaviour to new circumstances). But a unified explanation of learning has not been found, on the contrary, many controversies have arisen.

There is the "nature-nurture" debate. How much knowledge is born into a man or an animal and how much of it is completely new and acquired? The one extreme has it that all knowledge is already present at birth and that a learning process only activates the innate knowledge. This was the point of view of Plato. It is also called the nativist or rationalist viewpoint. The other extreme has it that man is born completely void of knowledge and that all of it is acquired. This is also called the empiricist viewpoint.

These extremes are not supported anymore, as the truth is somewhere in the middle. But a theoretical issue remains how much knowledge should be present at the beginning for a system to be able to learn at all, or what kind of ability to learn is needed.

Another problem is how we should study learning behaviour. This is done from two points of view, both of which represent major schools of thought in psychology. These are the behaviourist school, which considers a learning system as a black box. Only the behaviour of the system under different circumstances and in different environments is studied. The other school is called cognitivism. Here one *is* interested in what happens inside the system under study. One tries to build models of how the system works. The importance of ones point of view and ones school of thought is that ones definition of learning depends on it.

One could say that the study of learning computer systems is by definition cognitive because we try to model the inside of a learning system. However the measurement of its performance should be done from a behaviourist point of view. Moreover, there are learning systems, for example neural networks, that have such complex inner workings that they can only be evaluated in a behaviourist way.

I will now give a behaviourist as well as a cognitive definition of learning. The behaviourist definition is: "Learning is the ability of systems to improve their responses based on past experiences" footnoteKumpati Narendra, M.A.L. Thathachar, *Learning automata, an introduction*, Prentice Hall International, Englewood Cliffs NJ, 1989.. Note that this definition does not cover everything. It has difficulties with habituation and with acquisition of completely new abilities.

A cognitive definition would be: "Learning processes include the acquisition of new declarative knowledge, the development of motor and cognitive skills through instruction and practice, the organization of new knowledge into general, effective representations, and the discovery of new facts and theories through observation and experimentation"[13]. This definition does not really cover the more low-level learning phenomena.

It is very difficult to give a good definition of learning. In the rest of this lecture I will trust the listener to combine his intuition with the things I have just mentioned to get an idea of what is learning and what is not.

3 Approaches to Machine Learning

I will now proceed to attempt a classification of the different kinds of learning and the different kinds of learning systems that exist. But first I will try to classify the different environments in which a learning system can operate, because the kind of environment is of great influence on the performance of a learning system.

3.1 Environments

First of all an environment can be complex or simple. I will characterize a complex environment as an environment that is impossible (or infeasible) to model completely. A cluttered desk is an example of a complex environment. A blocks world is an example of a simple environment.

Then an environment can either be changing or unchanging. In an unchanging environment, the same actions of the learning system may cause different responses from the environment. If our learning system is a walking robot that tries to explore its environment, then it will probably get a different response if it does a step forward when it is standing on an edge then if it does a step forward when standing on a plain surface.

Also an environment can be free of noise or noisy. If an environment is free of noise then it will give identical responses in identical situations. With situation I mean the state of the robot as well as the state of the environment. Note that a changing environment could easily be confused with a noisy environment. However, a changing, but noise free environment will always give the same

[13]Ryszard S. Michalski, Jaime G. Carbonell, Tom M. Mitchell: *An overview of machine learning* in Michalski, Carbonell, Mitchell, *Machine learning, an artificial intelligence approach*, Tioga publishing company, Palo Alto, CA, 1983, pp. 3-23

responses to equal states of the environment and the robot. A noisy environment could give different responses.

Finally there are environments that have time constraints and there are environments without time constraints. A learning system operating under time constraints (called real-time or on-line operation) will have to provide results in a certain time limit. If it is not able to do this, the results are much less valuable, or even detrimental.

I have tried to present the criteria of classification of the environments in such a way that it is clear that learning in some environments is more difficult than in others. In fact the real world is one of the most difficult environments for a learning system to operate in: complex, changing, noisy and with severe time constraints.

3.2 Learning Methods

There are many different kinds of learning. These kinds of learning can be distinguished on the basis of how the information that has to be learnt is presented to the system. We can then make a list of learning methods[14]. The methods will be presented in approximate order of the amount of supervision necessary to learn.

The first and simplest learning method is rote learning or direct implementation of knowledge. With this method knowledge is put directly into the system by the user. The system can only store and reproduce the things it learned. In fact this amounts to not much more than ordinary database operation.

The second method is more complex. It is simple from a machine learning point of view, but it can be very complex from an artificial intelligence point of view. For this system, the system has to be able to understand instructions it gets from a teacher, to store the new knowledge and to integrate it with what it already knows. This method is called learning from instruction and it is quite similar to the way we learn things at school.

The third is called learning from analogy. When a learning system is using this method, it already has to have some knowledge about things that are similar to the task it has to learn. It has to change this knowledge into forms that it can use to solve the learning task. It then stores this modified knowledge as new, acquired knowledge and uses this to accomplish the task.

These first three methods are more or less symbolic methods. The system needs to have a well defined model of the world in order to be able to acquire the new knowledge. The next method, which is called learning by examples, can be implemented without an explicit world model. In this method examples are presented to the learning system, together with some extra information about these examples. This information could be whether they are positive or negative examples of a category, or to which category the examples belong. The learning system eventually learns to reproduce the extra information. A possible variation on this method is to let the system guess the extra information first and then tell it how good the guess was. This is called *reinforcement learning.*

The last, and most autonomous learning method is learning from discovery. This is unsupervised learning, as a teacher is not necessary. The system autonomously explores the world, builds and tests theories and keeps the theories that have proven useful. This method is comparable to the way science works.

We have now classified learning systems to the environments in which they operate and to the way in which they are trained. A third, and very important

[14]Based in part on: Ryszard S. Michalski, Jaime G. Carbonell, Tom M. Mitchell: *An overview of machine learning* in Michalski, Carbonell, Mitchell, *Machine learning, an artificial intelligence approach*, Tioga publishing company, Palo Alto, CA, 1983, pp. 3-23

classification criterium is the algorithm the learning system uses.

4 Learning Algorithms

Learning algorithms come in many different flavours. One can imagine that a learning system that learns by analogy is using an entirely different algorithm than a system that learns with reinforcement. As mentioned before, some learning methods imply a symbolic approach, whereas others imply a sub-symbolic, dynamic approach.

This is the main distinction one can make between learning algorithms, symbolic and sub-symbolic. Symbolic systems learn quickly and with high accuracy in specialized, relatively simple environments. Sub-symbolic systems learn less quickly and with less precision, but they remain efficient in complex and noisy environments, where symbolic systems break down.

Because of my research into autonomous agents, my main focus will be on sub-symbolic learning systems. These systems are most suitable for the environment in which an autonomous agent has to operate. But I will say a few things on symbolic learning systems first.

Purely symbolic learning algorithms use classical AI techniques[15], like logic reasoning and heuristic search of large search spaces to build and maintain an explicit model of their environment. The model is usually some kind of semantic network. Using this model, they can plan their actions. Impressive results in the areas like the discovery of mathematical theorems, the playing of games and the storing and understanding knowledge about simple domains have been achieved.

Another class of what could be called symbolic learning methods, are learning methods that use statistical information to construct a model of their environment. The resulting world models are less complex then in the purely symbolic approaches, and usually consist of a finite-state machine or a classification tree. The use of statistical information makes these methods more immune to noise than the purely symbolic techniques, but I will call them symbolic nevertheless, because the resulting model still has a reasonable one-to-one correspondence with the environment. These systems have been used in classifying complex domains and learning in noisy and dynamic environments.

The representations in sub-symbolic systems do not have a one-to-one correspondence with the environment. The knowledge about the world is usually *distributed* in the system. This means that one data structure stores data about multiple things in the environment and many data structures store information about a single thing in the environment. The interaction between the data structures determines the behaviour of the system. The data structures can for example be weights of connections in a network or extremely simple rules in a kind of rule base.

Learning in these systems takes place on a very low level by simple manipulations on the basic data structures. The learning system usually does not need any specific knowledge about the environment. Because of their distributed nature, these systems are immune to noise and can cope more easily with complexity. On the other hand, their distributed nature makes it much more difficult to guarantee good performance and makes it much harder to understand exactly what the system has learned.

[15] For an excellent overview of symbolic AI-learning (as well as a lot of sub-symbolic learning) see the excellent 3 volume series of Michalski, Carbonell, Mitchell: *Machine learning, an artificial intelligence Approach*

This brings me to the criteria against which learning systems can be evaluated. These criteria are important in determining if a certain learning algorithm is interesting or not. These criteria include: speed of learning, the number of different functions that the system can learn or the different environments in which the system can operate, the maximum complexity that the system is able to cope with. Also whether the system is tolerant to noise or not and the accuracy with which learning tasks are performed. Another very important criterium in learning systems is the ability of a system to *generalize*. Generalization is the ability of the system to cope with situations it has not seen before. Two other, more theoretic criteria are whether the system can be proven mathematically to work and whether the knowledge it has learned can be understood and explained.

Of course the importance of all these criteria is dependent on the learning task at hand. Therefore it is not possible to say that one learning algorithm, or even a class of learning algorithms is better than any other. It is only possible to assert that a certain learning algorithm is more suited to a certain situation than another.

5 Some Example Systems

I will now briefly explain how some examples of sub-symbolic learning systems work. I will start with the neural network.

Neural networks[16] [17] are learning systems that are based on how we think that real neurons work. Real neurons are connected with each other through axons and dendrites. If a neuron is agitated sufficiently at its dendrites, it will *fire*, that is, pass a signal through its axon. Learning is implemented by physical changes in the connections between one neuron's axon and another neuron's dendrite.

All this is modeled in an artificial neural network. The artificial neurons are the nodes in a network, the axons and dendrites are modeled as the connections in the network. These connections are connected to one neuron's output and another neuron's input. With them a *weight* is associated. The sum of the outputs of the neurons connected to the incoming connections multiplied with the weight of these connections is taken as the stimulation of an artificial neuron. Its output is then calculated using a certain activation function.

The network itself can be organized in many ways. Some paradigms use networks that have only symmetrical connections. Others have a network with distinct layers, where nodes in one layer are only connected with nodes in the next layer. All networks have input nodes and output nodes. Input nodes receive input from the outside world and the outputs of output nodes is taken as the output of the system.

If one uses a non-linear activation function, and a network that has a certain complexity, then one can approach all possible functions of the input to the output with any accuracy.

Learning in artificial neural networks is done by modifying the weights of the connections. Many different learning rules exist. The most well known are the

[16] For classical articles on neural networks: James A. Anderson, Edward Rosenfeld (eds.) *Neurocomputing foundations of research*, MIT Press, Cambridge MA, 1988

[17] For more theoretical background, see J. Hertz, A. Krogh and R.G. Palmer, *Introduction to the Theory of neural Computation*, Addison Wesley, Reading MA, 1991

Hebb-rule[18] [19] and back-propagation[20]. When using the Hebb-rule, connections are strengthened if the activations of the two nodes it connects have equal sign and weakened if they are unequal. When using back-propagation one calculates the discrepancy between the output a network produces and the desired output and updates all the connection strengths accordingly. The network must be organized in a feedforward fashion.

Artificial neural networks are the most well known and well researched subsymbolic learning system. They are used in a wide variety of applications, ranging from recognition of cancerous liver cells from microscopic images to error control in data communication.

Another well known and naturally inspired method is that of the genetic algorithm[21] [22] Genetic algorithms are based on evolution as seen in nature. They are considered a very powerful search technique, but they can also be applied to machine learning.

Genetic algorithms have some distinguishing features. They use a population of points in the search space, instead of the single point that is used in most search techniques. They usually use an encoding of the search parameters instead of the parameters themselves. This will reduce the search space. Furthermore they do not use any knowledge about the search space and they use a certain degree of randomness.

The genetic search is started with a completely random population. This population consists of a large number of individuals that are encodings of possible solutions. Each of these candidate solutions is then evaluated and assigned a certain fitness. The fittest individuals are then allowed to procreate. This can be done by crossing them with other individuals with high fitness or by mutating them slightly. Because only individuals with high fitness are allowed to procreate, the overall fitness of the population is increased and a good solution is usually found very quickly.

Genetic algorithms are very fast search mechanisms. Their only disadvantage is that they usually do not find the optimal solution, but only a very good solution. However, this does not really matter in most applications. One can easily imagine some possibilities to use genetic algorithms in machine learning. One of the best known is the classifier system.

Classifier systems[23] [24] are based on very simple rules, called classifiers. These classifiers have one or more conditions, an action and a strength. A classifier system contains many classifiers. Classifier systems also have a so called message list. Inputs to a classifier system are coded as messages and placed on the message list. These messages are then compared to the conditions of all classifiers in the system. Classifiers whose condition is satisfied are allowed to try

[18] Donald O. Hebb, *The Organization of Behavior*, Wiley, New York, 1949

[19] N. Rochester, J.H. Holland, L.H. Haibt and W. L. Duda, *Tests on a cell assembly theory of the action of the brain, using a large digital computer*, IRE Transactions on Information Theory, IT-2:80-93

[20] D.E. Rumelhart, G.E. Hinton, R.J. Williams, *Learning internal representations by error propagation*, Parallel DIstributed Processing: Explorations in the Microstructures of Cognition, Vol 1, D.E. Rumelhart, J.L. McClelland (eds.), MIT Press, Cambridge, MA, pp. 318-362

[21] John H. Holland, *Adaptation in Natural and Artificial Systems*, the University of Michigan Press, Ann Arbor, 1975

[22] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading MA, 1989

[23] J.H. Holland, *Escaping Brittleness, The possibilities of General Purpose Learning Algorithms Applied to Parallel Rule-Based Systems*, in R.S. Michalski, J.G. Carbonell and T.M. Mitchell (eds.), *Machine Learning, an artificial intelligence approach, volume 2*, Morgan Kaufmann, Los Altos CA, 1986

[24] For more modern research, see: *Evolutionary Computation* Special issue: Classifier Systems, (MIT Press Journals), volume 2, number 1, spring 1994

to perform their actions. But only the strongest classifiers have a good chance to succeed. Weaker classifiers have only a very small chance to perform their actions.

Actions can consist of many things, but they usually also cause the placement of a new message on the message list and sometimes the execution of an action in the outside world. The placement of messages on a new message list ensures that the classifier system can start a new cycle.

Meanwhile learning takes place. Classifiers that perform an action are rewarded or punished. Reward consists of an increase in strength, so that the chance that they can perform their action a next time increases. Punishment consists of a decrease in strength. Many different schemes to assign credit to classifiers exist. One is the bucket brigade algorithm, in which classifiers get rewarded by the environment and by other classifiers they activate. Another is Q-learning, a learning method in which an estimate of the best action at a certain time is calculated using the history of rewards and punishments of the system.

Thus far, the learning system is in fact a kind of symbolic learning using very simple rules. But a genetic algorithm is used to search for new rules, using rules as individuals and the whole classifier system as the population. The strength of rules is used to calculate their fitness. Special measures must be taken to preserve diversity in the population, as ordinary genetic algorithms would try to find one optimal rule, where in classifier systems there is usually no single rule that solves the learning task.

The combination of a genetic algorithm and of a group of very simple rules without domain specific knowledge that operate in parallel makes this approach a sub-symbolic rather than a symbolic learning algorithm.

Classifier systems are very good in solving simple, noisy stimulus response tasks and have good generalizing capabilities. However, their application to more complex domains has been problematic, although it is expected that they are able to perform quite well in these.

6 Learning System Used on an Autonomous Robot

In the research that I have done at the artificial intelligence laboratory of the free university of Brussels I have investigated a special kind of low level learning algorithm for autonomous robots. This learning algorithm has been invented by Luc Steels[25] but he has only researched it in simulation. My research consisted of implementing it on a real robot and of researching some variations on Steels' suggestion.

But first, let me say a few things on learning systems on autonomous robots in general. Learning systems have been part of robotics since the very beginning. Some of the earliest examples of robotic learning can be found in automatic drilling machines and automatic lathes. These could repeat a series of movements that had been performed on them once. The learning method was direct implementation of knowledge and therefore very simple and very inflexible. However, the usefulness of learning systems in robotics has been recognized from the very beginning.

More modern and less practical research has usually focused on the learning

[25] Luc Steels, *Emergent functionality in robotic agents through on-line evolution*, Proceedings of Alife IV, MIT Press, Cambridge MA, 1994

of higher-level skills, like map building, navigation[26] and manipulation. Lower level motor skills have also been investigated, for example the pole balancing problem[27].

Many experiments were only done in simulation, however. Very few researchers take the effort to really build a robot and implement the learning algorithm on it. Building a robot might seem unnecessary when one can do a simulation, but usually the environment of the robot will be of such complexity that even in the simplest experiments a simulation is also a simplification. Sometimes this will cause a real system to fail where a simulated system apparently succeeds.

Now let us see what the requirements of a learning system on a real robot are. A robot is supposed to work in the real world, so the learning system has to be able to cope with noisy and changing environments that also have time constraints. If the learning task is more involved, the learning system also has to be able to cope with complex environments. Furthermore, the system has to be able to learn on-line and without supervision. This limits the possible learning systems to reinforcement learning (where the system calculates its own reinforcement) and learning from discovery.

Then the system also has to work in real time and it should also not suffer from overtraining. Overtraining occurs when a system paralyzes because it has adapted too much to the training information. It is then not able to adapt to new circumstances or to generalize anymore. Overtraining is especially a risk to learning systems in autonomous robots, because the learning system has to be operating all the time, as one never knows when something worth learning happens. Furthermore a robot can operate comfortably in an environment for a long time without the necessity to do anything new, but it should remain to be able to adapt to a sudden change in the situation.

These requirements are very strict and I do not know of any learning algorithm that fulfills all of them. Fortunately not all criteria have to be met in all learning situations on robots. However, there are not many learning situations that are more difficult than the one on a robot.

The learning situation that I have researched was rather simple from the point of view of complexity. A wheeled robot had to learn low-level relations between its sensors and its actuators. In other words: it had to learn how its body functioned. For such basic things a learning algorithm is not usually used. Almost always low level reflexes are implemented directly. This has the advantage of accuracy, but a disadvantage is that a small change in the sensors or the actuators of the robot could cause the system to cease to function properly. If a learning algorithm is present, the system could adapt itself to the new situation.

The learning system I have used was based on PDL[28]. In PDL a lot of processes operate in parallel. These processes try to influence quantities by proposing to add a value to them. The processes in my research were of the following extremely simple form: $q_j \leftarrow \alpha(\gamma - q_i)$, where α is the strength of a certain process and γ is the goalvalue of a process.

[26] See for example: Dave Cliff, Inman Harvey, Phil Husbands, *Explorations in Evolutionary Robotics*, Adaptive Behavior, Vol 2, No 1, MIT Press, 1993, pp 73-110

[27] The pole balancing problem has been studied by many different researchers. The first to use a learning algorithm were probably: D. Michie and R.A. Chambers *Boxes: An Experiment in Adaptive Control*, in Machine Learning 2, Oliver & Boyd, Edinburgh, 1968 pp. 137-152
It has also been studied at the AI-lab of the Free University of Brussels: Bernard Manderinck, *Selectionism as a basis of categorization and adaptive behavior*, Technical Report 91-1 (phD Thesis), 1991

[28] Luc Steels, *The artificial life roots of artificial intelligence*, Artificial Life Journal, Vol 1,1 MIT Press, Cambridge MA, 1994

When the system is first started, these processes are generated randomly. They are then ran for a while, after which they are evaluated and their strengths are updated. Before the evaluation of the processes, a satisfaction of the system is calculated. This satisfaction is a real number and it is high if the performance of the system is good. Now the processes are evaluated according to their correlation with the satisfaction. If the satisfaction has increased, then processes that have proposed changes in the directions that the quantities have actually changed, are strengthened. If the satisfaction has decreased then processes that have proposed changes in the opposite direction of actual change of the quantities, are increased. The idea behind this is that the changes in the quantities were probably responsible for the increased satisfaction. The processes that were responsible for the changes are thus rewarded.

Sometimes a random new process is inserted into the population. There is a maximum strength for processes to prevent domination of the processes by a single one. Also the strength of every process is decreased by a certain percentage every time they are updated.

This learning system is quite comparable to a genetic algorithm, except for the fact that it does not use sexual multiplication and that it does not use an encoding of the search parameters, but the search parameters themselves. However it is possible that the system could benefit from the results from genetic algorithms research.

The system as it is now was found to be quite capable of learning simple relations between sensors and actuators. However, its computational strength was not sufficient to detect more complex relations, especially not when these involved derivates of- or integrals over sensor values. Currently I am working on some proposals to extend the computational strength. The system was also found to suffer from overtraining in some cases. By proper adjustment of the maximumstrength of processes and of the decay of process strength this problem was solved, but no satisfying theoretical explanation was found, something which will be necessary if a system is to be used in real applications.

This learning system is one of the first attempts to create a learning system especially for autonomous robots. I think that there is much opportunity for learning systems in autonomous robots, but there is also a lot of opportunity for them in all kinds of other applications of computers, ranging from embedded applications to graphical user interfaces to telecommunications to autonomous software agents. Learning systems may not be suitable for all situations, but a lot of the rigidity that appears to be a trademark of present day computer applications could be reduced with them. In this no single learning system will prove to be the best. All have their strengths and weaknesses. What systems are best applicable in what situations still has to be found out. And for some situations a good learning system still has to be found. A lot of questions are still waiting to be solved in learning computer systems research.