Introduction to Intelligent Agents

Contributors:

Frederick Mills: Author
Robert Stufflebeam: Author

Additional Credits:

Funding
This module was supported by *National Science Foundation Grant* #0127561.

What is an intelligent agent?

Generally speaking, the aim of cognitive science is to understand the nature and workings of intelligent systems. An intelligent system is something that processes internal information in order to do something purposeful. A great many things satisfy that description: People, computers, robots, cats, sensory systems, the list is endless. One sort of intelligent system of particular interest to cognitive scientists is that of an artificial autonomous intelligent agent.

But what are they? Well, let's break the term down. An agent is anything that is capable of acting upon information it perceives. An intelligent agent is an agent capable of making decisions about how it acts based on experience. An autonomous intelligent agent is an intelligent agent that is free to choose between different actions. As the term 'artificial' suggests, the sort of autonomous intelligent agents of interest to us here are not the sorts of things we find in the wild. Rather, they are created. Hence, an artificial autonomous intelligent agent is anything we create that is capable of actions based on information it perceives, its own experiences, and its own decisions about which action it performs. Since 'artificial autonomous intelligent agent' is quite a mouthful, let's follow the convention of using 'intelligent agent' or 'autonomous agent' for short.

Whether you are surfing the internet, shopping online, seeking a medical diagnosis, arranging for the transport of some commodity, or planning a space mission to explore the asteroid belt, intelligent agents are likely to play a key role in the process. These agents seize the initiative to seek the best plan of action to accomplish their assigned goals in light of the current situation and past experience, and then act on their environments. Intelligent agents are today being applied in a variety of areas, including: Distributed Project Management, Electronic Commerce, Information Retrieval, Medical field, Military, Manufacturing, Networking, Planning and Scheduling, and NASA Missions.

What disciplines support research and implementation of these special types of programs? The field of computer agent theory combines research in cognitive science and artificial intelligence.

Cognitive science provides the information processing model of mental processes used to describe the rational behavior of these agents in functional terms. Functional terms basically focus on what an agent does in pursuit of its goals. For example, a web search agent may have the goal of obtaining web site addresses that would best match the query or history of queries made by a customer. It could operate in the background and deliver recommendations to the

customer on a weekly basis. So its function is to map a history of queries to recommended web sites and deliver these recommendations via email or some other means of transmission. Diagrams can be used to represent the different modules that show how the agent receives input (web site visit history of customer), interacts with the environment (e.g. the internet), determines recommended web sites (decision procedure), communicates with other agents or data bases (communication protocol), and acts on the environment (using its effectors (something that produces an effect), e.g., transmission of email).

The field of artificial intelligence provides the technical skills for translating desired types of agent behaviors into programming language, related software, and the appropriate architecture (hardware and related software) for implementing the agent in a real or simulated world.

Since we are concerned first with the basic idea of agents, we begin by focusing on some basic concepts employed in agent theory. The aim of agent theory is to define and understand the distinguishing features of computer agents. Again, following what is now a convention, we will refer to such computer programs as intelligent or autonomous agents.

In order to understand how such programs are different from other software programs, we will begin by defining intelligent agents and then articulate this definition in more detail.

In order to get started, we first need a basic understanding of a generic agent, of which intelligent agents are just one type. Then we can begin to specialize later.

Generic agent

An agent is anything that perceives an environment through sensors and acts upon it through effectors (see Russel and Norvig, p. 31). This sounds simple enough.

This definition of agent covers a broad spectrum of machines, from thermostats (which do not learn anything new) to worms (which can actually learn a small repertoire of behaviors) to humans (with the greatest learning capacity, so far, on earth).

How does an agent perceive an environment? Sensors are the instruments employed by the agent to gather information about its world. A keyboard and a video camera can function as sensors if they are linked to an agent program. At the response end of the system, effectors are the instruments used by the agent to act on its environment. A monitor, a printer, and a robotic arm are examples of effectors. Let us look at the simplest type of agent, one that has a single mission, the thermostat.

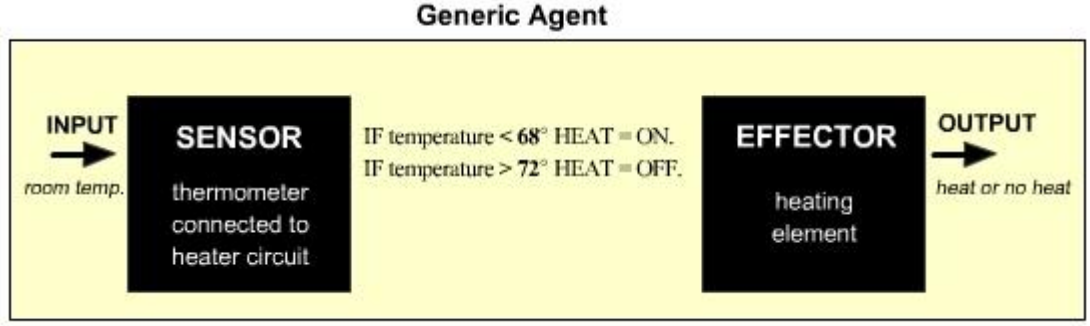

Fig. 1: Thermostat agent

Notice that the thermostat agent responds to a very specific feature of the environment with only three possible actions: turn heat on or turn heat off or take no action. Such an agent does not really qualify as intelligent or autonomous, for as we shall see shortly, its repertoire of behaviors and lack of flexibility and adaptability are just too limited to qualify as autonomous by definition.

Notice that even with a simple thermostat the environment determines the action of the agent and the agent's action, in turn, modifies the environment, in a relationship of mutual determination. When the temperature falls to 68 degrees Fahrenheit, the heat is turned on, but this brings the ambient temperature of the room to 80 degrees, which triggers an action to turn the heater off. Mechanical governors on steam and other types of engines function in a similar fashion.

The environment is generally the domain or world of the agent. These domains, at least for now, must be limited to specific types of situations in order to avoid the unlimited possibilities of the every day world. It is useful to distinguish two types of environments that impact on the computational challenges of agent programs.

If the environment is effectively accessible, the agent's sensors give it complete information about the state of affairs that are relevant to the agent's goals. The thermostat always has complete access to the room temperature and does not have to store any information. Its model of the world is really the world itself. Since such agents have access to whatever knowledge they need at any time, there is no need to store the state of affairs internally. A motion detector need not store information about objects because at any given time, all the relevant information is available to the sensors. It is when relevant information is not effectively accessible that an agent may need to store information and be equipped with a priori knowledge about certain features of its environment.

If the environment is deterministic, the future state of affairs is deducible from the current state of affairs; nothing is left to chance. Board games have this feature, even though the trees of possibilities may extend out into the billions of possible moves and counter-moves. Of course a non-deterministic world is in principle unpredictable. Different environments pose different challenges to agent designers.

As we increase the complexity of the environment and the variety of problems that must be solved in order for the agent to attain its goals, more flexibility and adaptability is required in the problem solving computational processes of the agent.

We are now prepared to examine a more refined definition of intelligent or autonomous agent, one that will better account for its distinguishing features.

Autonomous agent

Notice how the following definition includes the impact of the agent's own current behaviors on its own future behaviors.

An autonomous agent is "a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future" (Franklin and Graesser).

This definition captures all of the basic features of intelligent agents except their sociability. It provides a good approximation of the basic features of the large variety of intelligent agents now under development. Let us take a close look at this definition.

Intelligent agents sense their environments. We have seen this feature of agents in the first definition. But here the sensory data or percepts include not only data about other objects, but also about the impact of the agent itself on the state of affairs in the environment. The sensors could be organic, like eyes and ears and their neural processors, or artificial, like video and audio processors integrated into a digital computer. The environment could be a very limited domain, like a blocks world, or very sophisticated one, like the stock market or a set of asteroids. The sensors must be appropriate to the sort of objects with which the agent is designed to interact. Whatever the sensors, the history of all percepts sensed by the agent is critical to its future interaction with the environment.

The set of all percepts that an agent has to start with plus those gained by interaction with the environment (experience) is the percept sequence of the agent. An intelligent agent consults its percept sequence and the current state of affairs (which may be considered part of the total percept sequence) in light of its goals before taking action. This means that intelligent computer agents, like human agents, consult past experience and the current situation before deciding what course of action will further its goals.

In order to save computational power, it is possible to narrow down the search for relevant percepts by using short cuts. These short cuts, or heuristics, group percepts into classes of events so that the agent need not consult all classes of events, but only those that might assist in attaining its goals given the current state of affairs. If I am looking for two seats in a crowded movie theatre, I do not check every seat one at a time to see if it is empty. I use a simple heuristic: look for gaps between people's heads and check to see if those gaps indicate empty seats. This short cut will save me time so I do not miss the entire movie.

Now that we have some understanding of a percept sequence, we may ask: In what way does the agent's own actions form part of this sequence. The second definition states that the agent is considered a part of the environment. This means the agent inhabits a world or domain. It is situated. And it senses the impact of its own habitation. If I am waiting on line for a slice of pizza, my own behavior impacts on the overall length of the line. It may happen that because I have joined the line another cashier is called to work a second line. Thus my own impact on the world has changed the state of affairs to require a different sort of behavior from all agents behind me as well as myself. Another example is familiar to anyone who enjoys jumping in water. If I make a big splash in the pool, I then get to experience the series of ripples generated by my antecedent action. Another example comes from the medical field. If a drug dispensing agent has already administered half its prescribed dose to a patient, the amount already given alters the amount to be given in the future (unless there is a computational malfunction, in which case the tragic occurs). So when an agent acts on the environment, it senses the impact of its own acts, along with other events that fall within its domain.

The intelligent agent also reacts to its environment. Just as the generic agent has both sensor and effector, so too does the intelligent agent. But here things get a bit more interesting. A thermostat can react to its environment by turning the heater off when it ÒsensesÓ the air temperature reach the threshold of 72 degrees Fahrenheit. A thermostat, however, is not an intelligent agent. A mechanical thermostat does not even have a program. It is a dynamic system that is immediately related to the temperature of the environment. By complicating the

relation between sensing the environment and effecting the environment, we can build up our concept of agency to the level of intelligence and autonomy. To see this clearly, we will distinguish between a reflex agent, a goal-based agent, and a utility-based agent.

Reflex agent

A reflex agent is more complex then a mechanical thermostat. Rather than an immediate dynamic relationship to its environment, the reflex agent basically looks up what it should do in a list of rules. A reflex agent responds to a given percept with a pre-programmed response. Even if there are thousands of possible reactions to a given percept, the agent has a built in list of situation action rules to execute those reactions that have already been considered by the programmer. A situation action rule is basically a hypothetical imperative. If situation X is the current state of affairs and goal Z requires plan Y, than execute Y. Or even more simply, given X, execute Y. Thus for a medical diagnostic agent, if a certain set of symptoms is present, given a certain medical history, offer X diagnosis. Some expert systems fall under the category of reflex agent.

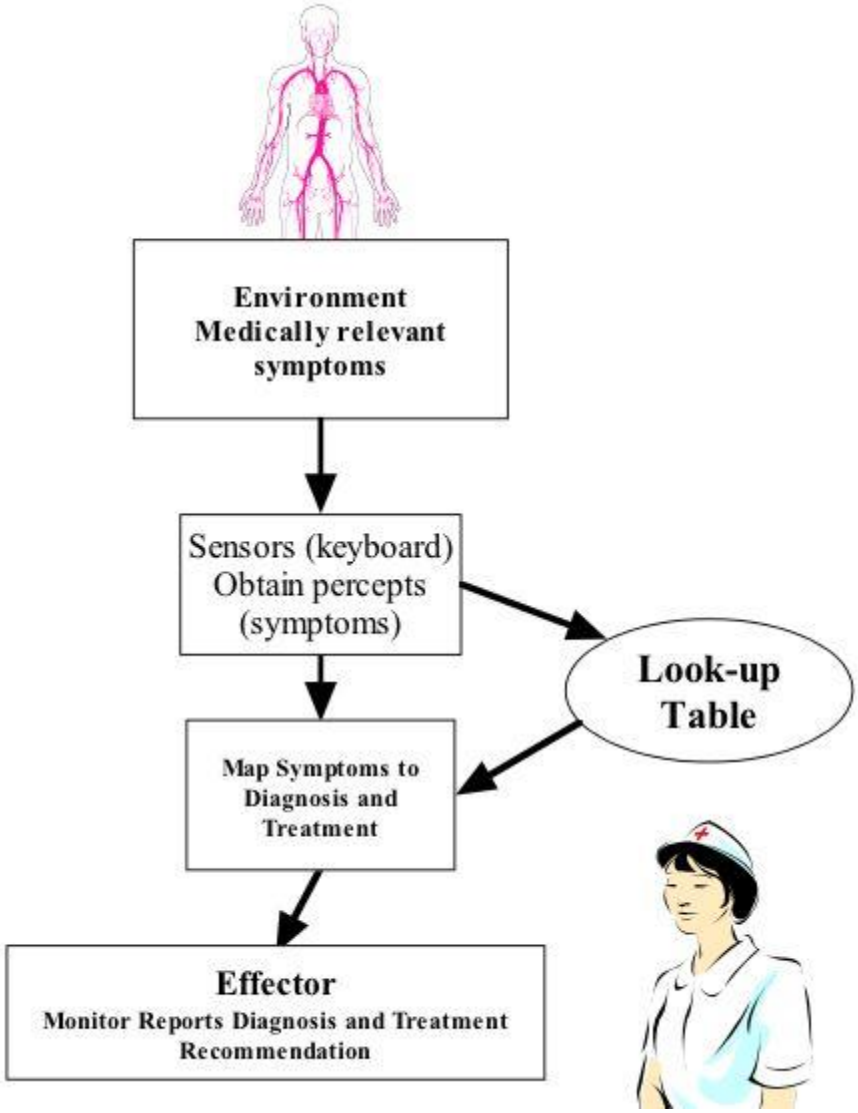

Fig. 2: Reflex Agent.

Reflex agents are really not very bright. They just cannot deal with novelty. If SARS is not in the database of pathogens, and the patient has symptoms associated with SARS, the reflex agent will not seek to update its records by consulting other agents. It will not accumulate experience that may indicate a new disease is present. It will not proactively check on patients to see if there may be some better diagnosis. It will not learn from its experience. If the percept is not in the reflex agent's database, the agent cannot react appropriately to the situation.

One might argue that the simple mercury type thermostat also is a reflex agent with only three rules. If the temperature reaches x, then turn the heater on. If the temperature reaches y, then turn the heater off. Otherwise, do nothing. The main difference is that the reflex agent requires a program that is not itself immediately and mechanically linked to the environment.

Goal based agent

An intelligent agent contains features of its less complex relatives, but it is not so limited. It acts in accordance with an agenda. It has a goal or set of goals that it actively pursues. A goal-based agent has a representation of the current state of the environment and how that environment generally works. It pursues basic policies or goals that may not be immediately attainable. Such agents do not ÒliveÓ merely in the moment as a thermostat does. These agents consider different scenarios before acting on their environments, to see which action will probably attain a goal. This consideration of different scenarios is called search and planning. It makes the agent proactive, not just reactive.

Another interesting feature of the goal-based agent is that it already has some model of how the objects in its environment usually behave, so it can perform searches and identify plans based on this knowledge about the world. Among the actions that can occur in the world are the agent's own actions. So the agent's own possible actions are among the factors that will determine possible future scenarios.

An intelligent agent called Remote Agent (we are interested only in its Mode Identification and Recovery feature, MIR) was used in 1999 to monitor the mechanical health of the Deep Space One NASA spacecraft, among other tasks. (Click for MORE INFO.) In a report on technology used in this mission, the software designers described the domain and concept as follows:

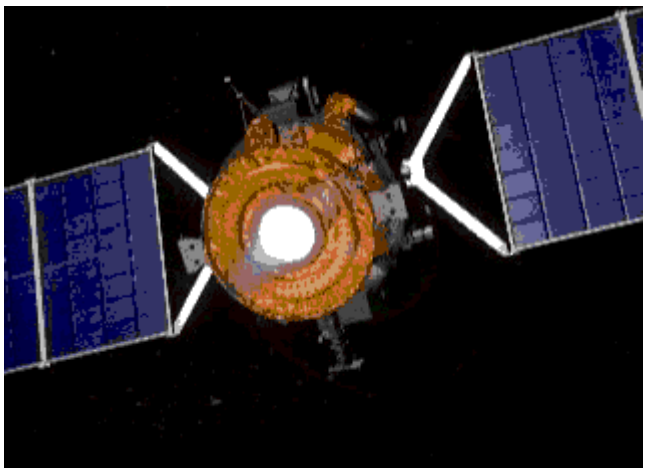

Fig. 3: Deep Space One (**NASA photo**)

DOMAIN:
Maintenance of internal spacecraft functioning, including automated vehicle health maintenance.

CONCEPT:
Model-based vehicle health management uses sensor data plus a stored model of spacecraft components and functions to reason about current state and to plan for recovery from system faults. The Livingstone fault diagnosis and recovery kernel proved its value in the successful Remote Agent Experiment (RAX) demonstration on Deep Space 1 (DS-1) in 1999.

Livingstone Model-based Health Management System, NASA Ames Research Center. Lee Brownston (QSS/ARC), James Kurien (PARC), Pandu Nayak (RIACS), David Smith (ARC/IC), Will Taylor (QSS/ARC). Group Lead: Mark Shirley (ARC/IC)

A health and safety agent on board a NASA spacecraft monitors the technical (not human) systems on board the spacecraft for any faults or signs of danger. If Remote Agent were a mere reflex agent, it would have to have a look up table for every possible scenario, which would take up too many computational resources and probably tax any engineer's imagination. As the agent developers of Remote Space pointed out in their report:

Software systems for vehicle monitoring, diagnosis, and control can be very complex, difficult to develop, difficult to debug and validate, and difficult or dangerous to modify during a mission. The software must allow for multiple simultaneous component failures -- possibly including sensor noise or failures -- and provide rapid response even to situations unforeseen [sic] by design engineers.

Remote Agent then, has the adaptability and flexibility to respond to novel faults in the technical systems on board the spacecraft. This does not mean Remote Agent was perfect. The behavior of even the ideal goal-based agent is not going to always be perfect with regard to attaining all of its goals. In the limited world of games, good moves can sometimes be deduced with certainty. But in the real life word we live in, nothing is certain, and operations on the environment only get us, at best, likely success. The same kind of limitations impact on the intelligent computer agents in analogous ways, even in the very limited domains in which, so far, they are designed to operate. (To see a simulation of Remote Agent in action, see http://ic.arc.nasa.gov/projects/remote-agent/applet/TimelineApplet.html)

Let us now look at a basic goal-based agent in more detail. In the following diagram notice how the agent's own actions modify the environment, thereby causing the agent to update its percept sequence.

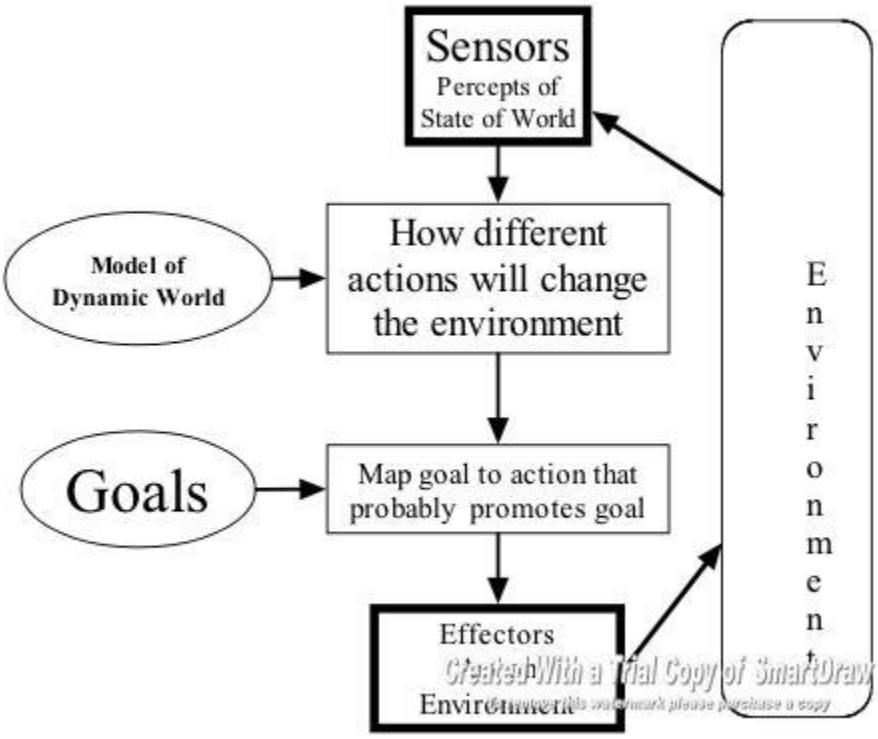

Fig. 4: Goal-based agent, adapted and simplified, from Russel and Norvig, p. 44.

Notice that the sensors proactively receive percepts to form a percept sequence. This percept sequence, combined with a model of the environment, brings about an updated model of the agents world. Now given the current state of the world and the goals of the agent, the agent can decide on an appropriate action to change its environment. The matching of the agent's percept sequence with an appropriate action is called mapping.

If the mapping procedure goes well, the percept sequence should lead to actions which further the goals of the agent, given the current state of affairs. This raises an interesting question about how the sensors gather percepts. How is it that relevant precepts are collected by the sensors and not just irrelevant garbage?

The autonomous agent is pro-active. Its percepts come about because of the manner in which it probes the environment. The sensors of the agent do not aim at just any data from the environment, but useful data, that is, percepts that may enhance its chances of performing future actions that promote its goals. Thus an agent that measures light waves from its environment will employ a spectrometer, not an oscilloscope, to probe its environment. In this sense, the agent effects what it will sense in the future, for it positions itself, by aiming at the right percepts, for future opportunities to achieve its goals. Of course the agent also generally effects what it will sense in the future merely by acting on its environment, for these very acts will bring about a new state of affairs that will itself be sensed by the agent.

Mapping raises a very important issue about autonomous agents versus other types of software. If a program is designed with a look up table that provides a specific action for all possible percepts of its world, we would have a program that always performs the right actions. The main problem with such a program is that it requires a very predictable world. Another problem is that, even if its world were predictable, it may be humanly impossible to create a table for all possible percepts. Even a chess game, with a limited number of pieces,

would generate millions of possible percepts, making it impractical to design a master level chess playing program with mere look up tables. This is why agent designers introduce flexibility and adaptability into their agents.

Flexibility is the capability to assess the probable outcomes to different possible actions as opposed to following one pre-scripted plan. Adaptability is the capability to make adjustments, including changing the current intention or action plan, given significant changes in the state of affairs, in order to promote the goal or goals. Both of these features rely on decision procedures to arrive at productive, rational decisions.

The two main fields of AI that study decision procedures based on percepts are Search and Planning. For our purposes here, the decision procedures used by the agent program determines the general manner in which the percept sequence leads to actions. It gives us the formula for mapping. I say general manner, because with intelligent agents, the programmer does not always know exactly what its progeny will do, given the large amount of computations that might involve just a few moments of activity.

How does mapping get to have some impact on the environment? The agent architecture is the machinery and software that runs the agent program. The architecture might be a Digital Computer with audio processors. Or it may be a robot with audio, visual, and even tactile sensors and programs that link the sensors to the agent program and the agent effectors. The effectors are the instruments the agent's uses to effect the world. This could be a simple email communication, the use of sophisticated scientific instruments, or a manipulation of objects in the world.

Utility-based agent

We will now add one more feature to our goal-based agent, to make it even more adaptive. In the more sophisticated agents, a utility measure is applied to the different possible actions that can be performed in the environment. This sophisticated planner is a utility-based agent. The utility-based agent will rate each scenario to see how well it achieves certain criteria with regard to the production of a good outcome. Things like the probability of success, the resources needed to execute the scenario, the importance of the goal to be achieved, the time it will take, might all be factored in to the utility function calculations.

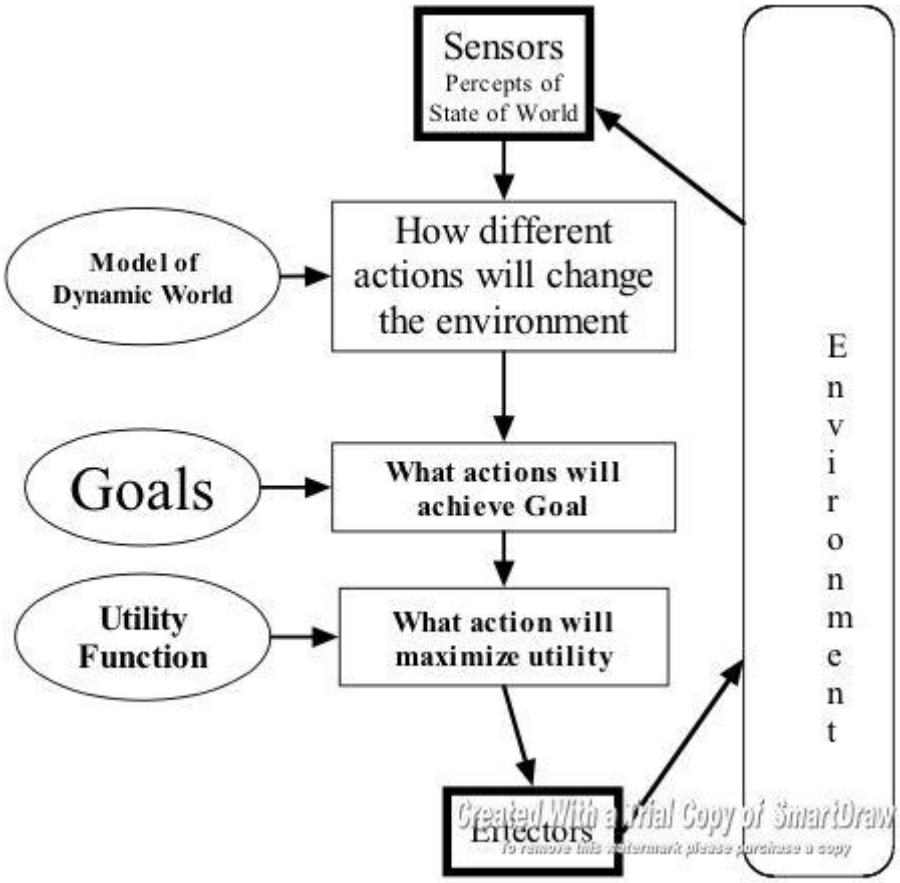

Fig. 5: Utility-based agent, adapted from Russel and Norvig, p. 45.

So far we have conceived of intelligent agents as rational utility maximizers that proactively pursue their goals. But what exactly makes them autonomous? They are autonomous, in part, because their behaviors are based not only on built in knowledge about their world, but also on their percept sequence. This percept sequence is not always predictable, especially in the case of non-deterministic, dynamic environments. It is the sensors of the agent which allow it to take input from the constantly changing world so that it can employ a decision procedure to map its percept sequence on to a plan of action in pursuit of its goals. Learning programs may also be implemented in the agent in order that the agent, through trial and error in novel situations, may still pursue its goals.

Since the programmer cannot generally predict every state of the world that will be confronted by the agent, the number of rules she would have to write for a reflex agent would be astronomical even in very simple domains, like scheduling meetings or arranging transportation routes and deliveries. But by giving the agent some goals, the ability to constantly reassess its situation, the ability to learn through trial and error, and in addition giving it a number of plans and ways of evaluating those plans as they become possible paths to the goal, the agent gets an enormous amount of flexibility and adaptability.

With our definition of intelligent agent in view, we are now prepared to examine the basic control loop written by agent theorist Michael Wooldridge (2000):

BASIC CONTROL LOOP OF AN AUTONOMOUS AGENT

while true

2. **observe the world;**
3. **update internal world model;**
4. **deliberate about what intention to achieve;**
5. **use means/ends reasoning to get a plan for the intention**
6. **execute the plan**

7. end while

This model needs some interpretation. The agent observes the world means the agent, using its sensors, collects percepts. A sensor may be the keyboard attached to a digital computer or a visual processor attached to a robot. It can be anything that enables the agent to collect percepts about the world.

Updating the internal world model means the agent adds the new percept to its percept sequence and pre-programmed information about the world. Such additions may alter the world model in small or large ways.

Deliberation about what intention to achieve, given the updated world model, is based on the overall goals of the agent. A new percept may alter the opportunities for achieving certain goals and call for a change of plans.

Once a decision is made about what intention to achieve, the agent consults its plan library and/or its decision procedures (e.g., means/ends reasoning) for determining what means to use to reach its end. Finally, the agent executes the plan, provided no new percept calls for an altering of its current intention.

If we add to this model the ability to learn from interacting with other agents, human and computer, we increase the flexibility and adaptability of the agent and add to its computational resources.

What is missing from the second definition of intelligent agent is the sociability of intelligent agents. Most agent theory now employs multi-agent system (MAS) design that employs a community of agents to pursue the goals. MAS helps to overcome resource limitations of isolated agents by having agents share information and complement each other's functions. (SEE MODULE 2 on MAS)

Taxonomy of agents

There is no consensus on how to classify agents. This is because there is no agreed upon taxonomy of agents. With this in mind, let us begin to classify the different types of agents, using some suggestions from the field of agent theory. Charles Pettrie, Stan Franklin, Art Glaesser, and other agent theorists, suggest that we provide an operational definition. So we will try to describe the agent's basic components and specify what the agent seeks to accomplish. Using the above definition as a guide, we specify an autonomous agent by describing its:

Environment (this must be a dynamic description, that is, a description of a state of affairs that changes over time as real life situations do).

Sensing capabilities (this depends on the sensor equipment; it determines the sort of data the agent is capable of receiving as input).

Actions (this would be a change in the environment brought about by the agent, requiring the agent to update its model of the world, which in turn may cause the agent to change its immediate intention).

Desires (these are the overall policies or goals of the agent).

Action Selection Architecture (the agent decides what to due next by consulting both its internal state, the state of the world, and its current goal; then it uses decision making procedures to select an action).