JOURNAL OF COMPUTER AND MATHEMATICAL SCIENCES
An International Open Free Access, Peer Reviewed Research Journal
www.compmath-journal.org

ISSN 0976-5727 (Print)
ISSN 2319-8133 (Online)
Abbr:J.Comp.&Math.Sci.
2014, Vol.5(2): Pg.209-217

Recreation of an Agent Based System Performance in an Active and Unforeseen Situation

P. Lokesh Kumar Reddy[1], B. Rama Bhupal Reddy[2] and S. Rama Krishna[3]

[1]RamaRaja Institute of Technology and Science,
Tirupati, A.P., INDIA.
[2]Department of Mathematics,
KSRM College of Engineering, Kadapa, A.P., INDIA.
[3]Department of Computer Science,
S.V. University, Tirupati, A.P., INDIA.

(Received on: March 28, 2014)

ABSTRACT

This work is focused on modeling an intelligent system agent based in dynamic environment. The model we propose is a software system in which an agent interacts in a dynamic, adaptive and evolving way. This computational approach allows us to overcome some difficulties that are present when we want to consider our system as an intelligent decision maker. In this way we are able to study different types of agent environments, and interpret them in the relevant effective simulation. Consequently we can observe how different situations of agent conditioned by environmental dynamism may lead the system to different behaviors. In particular, we are interested in studying the environment - agent interaction. For this purpose, we simulated both static and dynamic environment, to show the different behaviors of agent in the given cases for a larger base of discussion about the level of intelligence.

Keywords: Agent, Environment, Active, Planning off-line, Planning online, Agent architecture.

I. INTRODUCTION

The world is becoming more complex and an intelligent system needs to become more flexible in term of its interdependencies with the world. Today the computation requires new methods for architectures, with more autonomy.

Identifying agents and specifying their behaviors, representing agent environment interactions: these are some processes to start the work in developing useful agent models. Agents are generally the decision-makers in a system. Their nature can be different depending on many factors. In all cases, the first concept for an agent based on the system is: actions are done by the agent on the environment, which in turn provides percepts to the agent[1].

We see in their environment the main factor that conditions the agent behavior. Particularly, we are concentrated with analyzing and discussing the intelligent systems behavior in dynamic environment. The purpose is to combine the different behaviors getting the maximum of system performance. Our efforts are focused on making a two dimensional environment where the simulation will be developed: a static environment and dynamic environment. We have the possibility to design a model that incrementally will grow up the intelligent behavior of the system. The idea is a bounded virtual map where we have randomly set obstacles that cannot change their position during the simulation. The map is partially known. There is another obstacle we can set unexpectedly along the simulation. Even though we can modify the conditions, the agent should be able to gain new knowledge over the word and to reason about uncertain situations to achieve the goal. There is currently much interest concerned in this aspect of simulation of un-programmed movement, especially for applications where the movement is expected to be exploratory rather than purposive.

II. MODELING AN AGENT BASED SYSTEM

If we want to build an agent based system, first, we should identify the purpose of the system. In other terms we should specify the problem for which the system gives the solution. Next, systematically analyze the system under studying, identifying components and component interactions, relevant data sources, and so on. Then, apply the model and conduct a series of experiments by systematically varying parameters and assumptions.

There are some standard models of agent based simulation that require first of all, to identify the agents and guess a theory of agent behavior[2]. The general steps in building an agent model are:

1. Identify the agent types and other objects (classes) along with their attributes.
2. Define the environment where the agents will live in and interact with.
3. Specify the methods by which agent attributes are updated in response to either Agent-to-agent interactions or agent interactions with the environment.
4. Add the methods that check which agents interact, when do they interact and how they interact during the simulation.
5. Implement the agent model in computational software[3].

Agent definitions

About agent based systems, new concepts and techniques are being developed by many researchers. These efforts aim to make these systems more intelligent and flexible in real environment, as well as to improve the performance and safety on completing their tasks independently. From a practical modeling standpoint, there are many definitions where agents are considered to have certain characteristics. An agent is identifiable, a discrete individual with a set of characteristics and rules governing its behaviors and decision-making capability[4].

Agents are self-contained. The discreteness requirement implies that an agent has a boundary and one can easily determine whether something is part of an agent or not or is a shared characteristic. An agent is situated in an environment in which it interacts with other agents. Agents have protocols for interaction with other agents like those attributed to communication and the capacity to respond to the environment. Agents have the ability to recognize and distinguish the traits of other agents[5].

An agent may be goal-directed, having goals to achieve (not necessarily objectives to maximize) against its behaviors. This allows an agent to compare the outcome of its behavior connected to its goals. An agent is autonomous and self directed. An agent can independently function in its environment and in its dealings with other agents, at least over a limited range of situations that are of interest.

An agent is flexible. It has the ability to gain experience and apply it in future compartments. This requires some form of memory. An agent may have a rule that modifies its rules of behavior.

Another definition might be the agent is a computer agent able to make flexible actions in dynamic environment. Figure 1 shows a structure of an agent that acts in a real world. He executes his tasks changing the world and he needs to take new knowledge to update his model of the world.

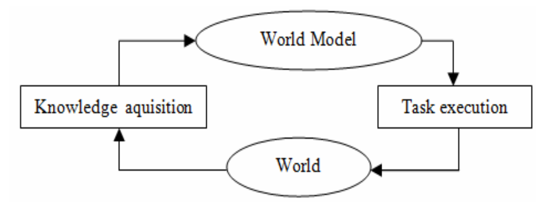

Figure 1. An agent structure

Sometimes there are other agents in the same environment. Now the agent concept is different. There is not a unique definition but it all converges to a common point: the agent is a special software component that secures a continuous interface with an agent system or human agent[6].

Searching the agent based system architecture

Starting from a set of requirements deducted in pre-phase of the analyze of the system, we need an intelligent system which can act in real world. Many uncertainly situations, not complete knowledge about the world versus hard requirements for rationality, safe decision making: this is a simple vision for our problem. There are some models of architectures for rational agents[7].

Deliberative architecture- representation of word in highest level, a set of actions very clear a cycle SPA (sense-plan-act) repeated and a control flow with a unique direction: sensors data, data interpretation modeling word, model reasoning, system control and system actuators.

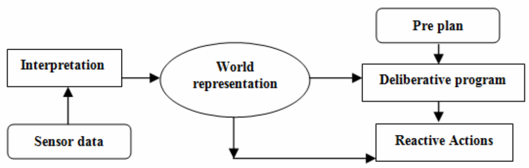

Figure 2. Deliberative architecture of an ABS

Reactive architecture - is led from principle intelligence is born from the interacting with environment. Reactivity is a process depended on the existence of input - output connections. The agent has inside this world with its rules and there are some modules where everyone takes a set of behaviors with certain function.

Figure 3. Reactive architecture of an ABS

There is not a model of real world and he must choose the right action in base of sensors information data. The main advantage of this architecture is the very fast reaction in unpredicted situations.

Hybrid architecture - has two levels, first deliberative level where the model of environment is taken and is made decision over the next action. And second the operative level where the conditions on the world are verified and then are made actions.

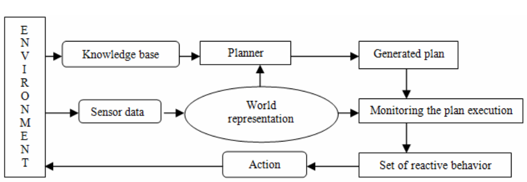

Figure 4. Hybrid architecture of ABS

This architecture implements an important phase which is planning in dynamic environment.

III. THE SIMULATION

This section has a goal to discuss the modeling process for simulation of a rational behavior of ABS. This will be done through a software application that we have constructed for this purpose. We represent here an application that represent, an agent which will operate in a two dimensional environment, with several obstacles positioned in the environment; the obstacles are generated randomly i.e. in random sizes and distance between them, and their shapes are circular.

Over this we can put another obstacle during the simulation time. A great number of obstacles give a complex virtual world but this required more calculation power managing the dynamic information.

Requirements of software system

We must design a software that simulates the navigation of an object called agent in an dynamic environment. Studying these conditions, we arrive at a question. What is the best solution? For us it is clear that our system must be intelligent. Going on, it must be based on the agent paradigm for the environment definition: dynamic and unpredicted (or unexpected).

The main problem of simulation is the agent navigation in a virtual non-real environment. While the robotic agents in the real word may be oriented through the sensors, here we have found the solution on the computational geometric methods for the object localization.

At each step, agent should performs two phases:

- interaction with environment if agent does not see the target,

• orientation choosing the next step.

The final objective is to find the target, executing a dynamic algorithm. For this purpose, we combine the two behaviors to achieve the goal. The agent should be able to control the environment, then to find the objective in real time and to make decision of reading this target in a satisfied interval without touching the information processing time that avoids every wrong path based on the old path planning. First, the agent aims at finishing the mission successfully (find target) and second to show a computed strategy that allows it to meet the target. The system is designed to control constantly the changes of the environment avoiding the failures.

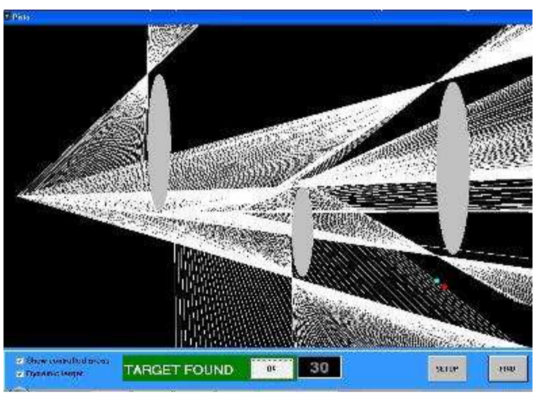

Figure 5. A general view of simulation

Analyzing the agent environment

We distinguish our agent operates in a not accessible environment because our agent does not detect all aspects that are relevant to the choice of his actions.

Another characteristic of the simulation environment is dynamism: we simulate two opposite environments:

1. Dynamic environment changes the target position of the environment while the agent is deliberating,
2. Static environment: the environment does not change with the passage of time.

Perception of environment

Our system is designed to have a visual sensor. What does it mean? The agent can percept the obstacle and can observe the distance from the obstacles. Hanging around the motion map, the agent can discover invisible areas behind the obstacles to find the target which is not in the visible area. Here he finds the motivation to change position for a new state with purpose to reach his objective.

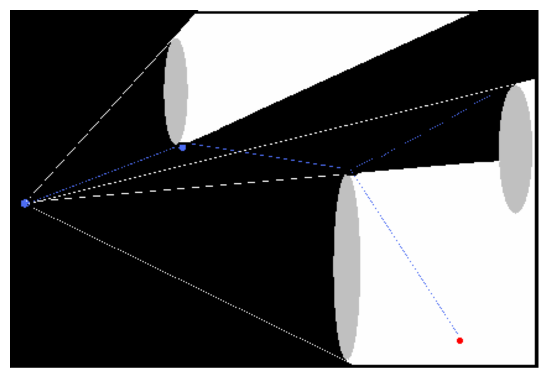

Figure 6. Agent moves following the visibility graph

The agent has not complete knowledge about environment. He starts his movement following a predicated plan in base of the visual sensor information. The graph visibility algorithm[8] is used with purpose. The agent chooses the shortest motion line and discovers the invisible area where the target can be located. He performs a deliberative behavior until we set an unpredicted obstacle on its navigation path. In this moment he shows a simple reflective behavior to solve the situation calculating a new path dynamically.

Static Environment

Preconditions: static target and static obstacles. To find the target, we model the agent behavior like a human behavior. First, agent has taken the task staying in two situations:

1. The target is in the visible areas (white areas in the Figure 6). He moves towards the target accomplishing the mission.
2. The target is in the invisible area (black areas). Agent should execute a new motion plan which consists of two actions:

a. Control the new visible areas to find out if the target is there.

b. Navigate performing an exploratory behavior. His immediate goal is to update the knowledge about his virtual world.

Dynamic Environment

Exploration of environment – the agent must see the target. If the target is out of the black areas, then the agent executes a part of algorithm that gives a short path to the target following the line between them, also it should control the invisible areas where the target can be located. The agent is prepared to stay in two situations predicted above.

Equipping the agent with visual sensor, he is to be able to see in real time the changes of environment while he taking his task. In the software design maximal capacities are required for a greater performance of agent.

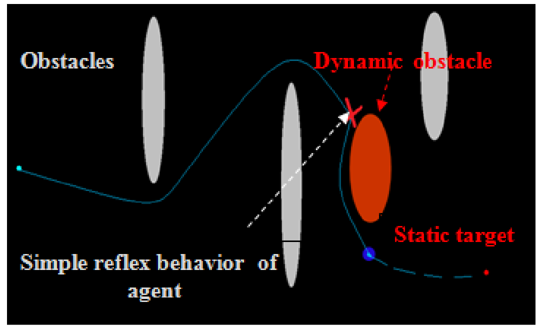

Figure 7. Agent performs an simple reflex behavior

IV. PATH PLANNING

Planning dynamically the next path, the agent must perform more rationality. In the dynamic environment the agent has a pre plan but this plan can change because environment map changes. The main task of a planner is to find a sequence of actions that allows it to complete some objectives. In dynamic environment we need a planner that makes a plan as a sequence of actions that bring the system from one state to another. This plan has a final state defined by the goal. One of the requirements for intelligent system is to generate an optimal plan from the own planner.

A decisive plan would not be useful in the dynamic environments. The reason is the uncertain information about the environment that changes during the work time of the system. Adaptation to the dynamism is one of the tasks of the planner which should make a dynamic plan with some properties[9].

The structure of modeling is based on a state machine model. We have used the hybrid model of architecture: deliberative agent with reactive planner shown in Figure 6. The plan is generated automatically from a reasoning tool based on case. There is a knowledge base at the start of simulation. A planning procedure is called that defines the next path from the generated plan. The agent program has a complex analyzing process that has to choose the right action depending on the sensor data. When the agent knows the perceptive sequence of the information along all the processes, it can execute a motion plan. This is the first case when the environment is static. The path is calculated by using simple methods of geometry (figure 8). However this is an ideal case because it is impossible to predict all the perceptions.

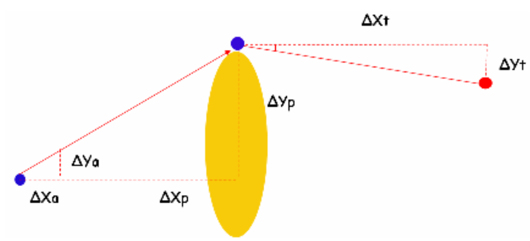

Figure 8. A simple geometric method to calculate the next step of the agent

At the same time there is a module that sees the virtual world and offers a map for the planner. The system controls the execution of the plans in sequential intervals making decisions over choosing the next path if there are changes in the environment. In this manner the system can manage uncertain situations generating specific plans. In this case the agent needs more compromises between stability and changes in adaption.

V. ALGORITHM OF AGENT MOTION

The representation of the actions needs some definitions as an essential condition for generating the action, then is needed a short description of the agent behavior and in the end other conditions to finish the action.

This is an important moment for showing the new state after the action execution. These actions are divided into two categories:

- normal actions that change the environment
- sensing actions that accumulate information over the environment to make decisions.

In intelligent systems during the execution the agent decides itself about the state of conditions: true or false.

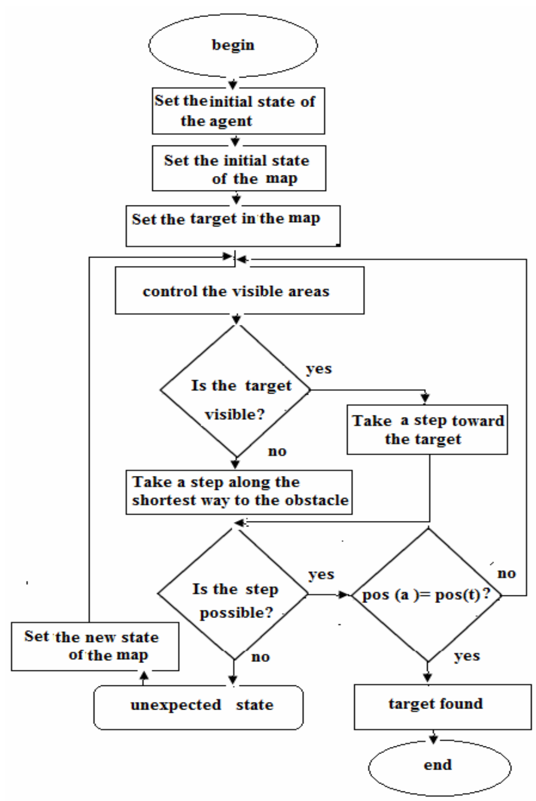

Figure 9. Algorithm of agent motion in unexpected environment

VI. RESULTS OF APPLICATION

During the simulation, we can observe the agent environment relations and their dependencies:

- We have designed a software system that is capable of integrating successfully the execution ability of complex tasks using reflexive capacities needed to manage uncertain situations in dynamic environment.
- The system architecture is hybrid that means we have chosen an architecture that implements reflexive and proactive agent behaviors.
  - When the environment is static and partially accessible, agent makes deliberative behavior.
  - When the environment is dynamic, his behavior is not constant because it is conditioned by the state of agent in the environment.
- We observe temptations for more reactive behaviors in expected situations.
- The speed of mission is another of agent exigencies. However the environment changes with certain speed and the agent has not time enough to make a perfect decision and to choose accurately the next action.
- Time in disposition does not compromise with the agent performance in the dynamic environment.

VII. CONCLUSIONS

The simulation software that we have designed provides a simple way to study the complex interactions among different types of environment and agents.

We were interested to study the stability on making decisions of the system. Our attention was focused on how much security offers an agent based system in difficult situations.

Many different types of agent behaviors have been introduced. We could observe the agent efforts to reach the goal and the results were interesting about agent intelligence.

In further research it would be interesting to introduce new types of environment making them more complex. We aim at extending agent based systems on modeling hierarchical structures of control systems.

REFERENCES

1. Wooldridge, M. and Jennings, N.R. A Survey, in Wooldridge and Jennings Eds., Intelligent Agents. Agent Theories, Architectures, and Languages, Springer-Verlag, Berlin, pp1-22 (1995).
2. Charles M. Macal, Michael J. North Macal, C. M., Tutorial on agent based modeling and simulation part 2 : how to model with agent , Simulation Conference, 2006. WSC 06. Proceedings of the Winter, Monterey, CA, Publication Date: 3-6 On page(s): 73-83 Dec. (2006).
3. Russell, S. and Norvig, P. Artificial Intelligence: A Modern Approach. Prentice–Hall, pp. 31-33 (1995).
4. George F. Luger, Artificial Intelligence Structures and Strategies for Complex Problem Solving Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA , pp. 245-272 (1997).
5. Michael Negnevitsky Artificial Intelligence A guide to Intelligent Systems,

Addison-Wesley, pp. 25-33 December (2004).

6. Keith Decker, Katia Sycara and Mike Williamson, Intelligent Adaptive Information Agents, *Journal of Intelligent Information Systems*, Vol. 9, pp 239-260 (1997).
7. Michael Wooldridge and Nicholas R. Jennings, Intelligent agents: Theory and practice, *Knowledge Engineering Review*, Vol. 10, pp115-152 (1995).
8. Penn, Alan and Turner, Alasdair, Space Syntax Based Agent Simulation, Springer-Verlag, Berlin, Germany, pp. 99-114 (2002).
9. Stuart J. Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, Pearson, pp.31-52 (2003).