Design Principles for Intelligent Agents in Distributed Interactive Simulation

Ercan Oztemel and Sakir Kocabas***
Tubitak- Marmara Research Center, Department of AI, P.K 21 Gebze, 41470 Kocaeli, TURKEY
** eomam1@yunus.mam.tubitak.gov.tr ** uckoca@cc.itu.edu.tr*

Abstract

This paper describes the basic design principles for intelligent agents in a distributed interactive simulation (DIS) environment. Agents in different simulation nodes in a DIS environment can use a common data or knowledge base as well as agent specific knowledge bases. Intelligent agents must be capable of performing their activities under real-time constraints. Such activities include: perceiving events, reasoning about events and responding to perceived events in an intelligent (and human-like) manner. Intelligent agents may have to face unexpected events and may have to rely on uncertain or incomplete information about their environment in deciding their actions. The design principles for DIS systems including intelligent agents operating in real time environments and scenarios are outlined in the paper. The paper also discusses these principles within the framework of an intelligent real time simulation project (EUCLID RTP 11.3) carried out by our group at Marmara Research Center (MRC)

1. INTRODUCTION

In recent years, there has been an increasing trend for using intelligent agents in simulation systems (see, [1,2,3,4,5,6]). Main purpose of these systems is to create more realistic environments for training certain skills and the analysis of events in certain scenarios. Realistic behavior in such systems increase confidence in them as mediums of training and analysis. On the other hand, training large number of people in a complex environment requires more than one simulation systems cooperating in a distributed environment, rather than using isolated simulation stations. This paper describes the basic design principles for intelligent agents and their cooperation in a distributed interactive simulation (DIS) environment. DIS is mainly concerned with time and space-coherent, synthetic representation of real world environments designed for linking the interactive activities of operational entities. The synthetic environment is created through real-time exchange of data units between distributed, and computationally autonomous simulation applications in the form of simulations, simulators and instrumented equipment interconnected through standard computer communicative services [7]. The computational entities may be present in one location or distributed geographically. It should be noted that DIS is generally employed for military training but with potential applications in nonmilitary problems such as crisis management and disaster handling. Dynamic events such as military training require a great deal of effort and large numbers of people involved. This makes the training of personnel for these events, very expensive and difficult to handle.This is why interactive simulation, in which computer generated entities are employed, is widely needed in such areas. Computer generated entities need to be intelligent so that they can deal with the complexity of the environment. Having these agents operating in a DIS environment will help to simulate dynamic and complex events in more realistic scenarios. The agents are intelligent software systems designed to satisfy internal or external goals by their own actions during continuous, long-term interaction with the environment in which they are operating [8]. Agents in different simulation stations can use the same database and share same knowledge base to perform their activities. They can communicate with each other in a realistic manner, and they operate under real-time constraints, perceive events, reason about them and perform required activities. They may respond to unexpected events using even uncertain or incomplete knowledge. In some cases (e.g. in mixed exercises) human agents such as armoured troops, in addition to pilots in manned simulators can take part in the training process. This makes the question of cooperation an important issue, among the elements of the simulation applications including computer generated entities and human counterparts. To achieve a certain level of cooperation, it is necessary to have an interactive simulation environment for communication. DIS environments provide such facilities. The next section examines the basic characteristics of DIS environments.

2. BASIC CHARACTERISTICS OF A DIS SYSTEM

The primary objective of a DIS environment is to create synthetic and virtual representations of the world to a degree that is sufficient for training. A DIS system has the following characteristics [9]:

. No central computer is used for event scheduling or conflict resolution.

* *Also affiliated with Sakarya University, Dept. of Industrial Engineering ADAPAZARI*
** *Also affiliated with ITU, Dept of Space Sciences and Technology, ISTANBUL*

. Autonomous simulation stations are responsible for maintaining the state of one or more simulation elements.

.There is a standard protocol for communicating ground-truth data.

. Receiving stations are responsible for determining what is to be perceived.

. Simulation stations communicate only changes in their state.

. "Dead reckoning" algorithms are used to reduce overloads in processing communication data.

These characteristics imply that several simulation stations can be connected together for the purpose of training (in some cases this figure exceeds hundreds). Standard protocols are used for data communication and interaction between the simulation stations and elements (see, references [7,9] for such data protocols). The extent to which a participant must comply with the DIS standards depends on the functions that the participant is expected to perform. If the participant is a computer generated intelligent element, then its behavior can be controlled by an intelligent control program either on the same station or on a parallel station using the same DIS network. Designing an intelligent agent requires a clear definition of its behavior in a scenario or exercise [9]. Intelligent agents consists mainly of three components: perception, cognition and action. Memory, reasoning, learning, understanding, planning, scheduling and control are the basic characteristics of intelligent behavior. An agent equipped with these capabilities can organize knowledge about the environment, interpret situations, deduce conclusions, solve problems, and create actions.

Designing intelligent agents requires a systematic analysis of the environment in which the agent is to operate. System architecture, knowledge acquisition, interaction, cooperation, communication, conflict resolution and learning are some of the other issues to consider. Such an agent, called AISim, is being developed at MRC within the framework of EUCLID RTP 11.3 battlefield simulation project. AISim is an intelligent agent designed to perform tasks in an air interception mission. Its tasks include take off, navigation, patrol, beyond visual range (BVR) and within visual range (WVR) engagement, air to air refueling, disengagement, and landing. The following procedure is employed in the design of AISim.

. Domain analysis (knowledge acquisition), to define the activities (functions) to be simulated in the simulation application.

. Requirements analysis, to define the system's goals and functions.

. Global design analysis, to ensure that each specified goal is achieved by a set of functions.

. Detailed design analysis, to guide the software engineer to code the AISim in accordance with the specified requirements.

. Software development which is the actual code generation process.

. Testing, verification, and integration.

It is important to define the goals of the agent as clearly as possible. Experiments have shown that a hierarchical approach would be suitable for the design of intelligent agents [5]. In its layered system architecture, AISim has four levels of goals:

1) mission goals
2) task goals
3) sub task goals, and
4) action goals.

Military operations require the definition of a mission goal such as air interception and tactical air support. Each mission is divided into a set of task goals such tactical air combat, patrol, navigation, and air to air refueling. These tasks are further divided into subtasks such as situation assessment, trajectory guidance, tactics management and evasion. The subtasks are in turn transformed into action goals such as firing a missile, increasing or decreasing altitude, and changing heading.

During the analysis stage of our work, a prototype called RSIM [1] was developed to help define the design specifications. The prototype development is important early in the design of intelligent agents, because, due to real time constraints, and the nature of knowledge employed, it would not be possible to consider every aspect of the design beforehand. The prototype is used to find out design variations and to test the feasibility of the AI techniques employed in the system. RSIM has proved to be useful in testing learning techniques.

3. GENERAL SYSTEM ARCHITECTURE

The basic hardware structure of the system on which AISim runs, is given in Figure 1.

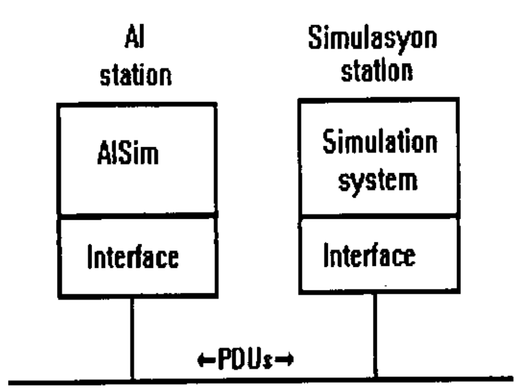

Figure 1:Hardware structure of an intelligent agent in DIS

The figure shows that AISim is to operate in a multi-agent environment. The design process of the program and the software development is still continuing. The system can now communicate with the simulation system (ITEMS) through an interface software (InterSIM) on a local network. AISim and InterSIM run on a Silicon Graphics workstation (IRIX 5.2) whereas ITEMS runs on a different workstation (Indigo 2 Extreme). AISim directly controls the behavior of an AI target, an F16, on the simulation workstation. Once AISim takes its final structure then it will be able to communicate with other agents in different simulation stations and manned simulators on a long haul DIS network.

Several software architectures for intelligent agents have been presented in the literature [1,2,3,5,10]. The main structural components of such systems and control flow are given in Figure 2. As seen from the figure, There are mainly 3 components in the architecture; perception, cognition and action. The perception component performs situation assessment by acquiring and interpreting sensory information of external events. The cognition component, on receiving the information provided by the perception component, performs reasoning and decides the activities of the agent in the dynamic environment in which it is operating. The action component is responsible for the actions to be taken with respect to the current situation. Some of the functional requirements of intelligent agents are as follows: dealing with dynamic objects, interpreting continuous and asynchronous sensory information, focusing attention to the important events in the environment, creating and modifying action plans, gracefully degrading the utility of the behavior, improving the behavior through learning, predicting future situations and events, prioritizing and scheduling the events under real time constraints. If such an agent is to operate in a DIS environment, it has to know how to cooperate and interact with other agents in the same or different simulation stations.

4. INTERACTION, COOPERATION, AND DATA COMMUNICATION

The cooperation of agents depends on the kind of tasks and activities they are expected to do, and the environment in which they are operating. There would be three possible cases[11]: Agents may perform problem solving in a common domain, agents may be working together to improve their individual performance, and agents may be working together to improve the performance of the overall system they are designed for. In DIS systems the third type of cooperation is important as it brings the question of dependency between agents. If an agent needs to communicate with other agents, it needs to know about the underlying model of these agents as well. There should be a global standard data communication which is accessible by every entity within the overall system. Some data communication problems can be solved by "dead reckoning" algorithms. Such algorithms estimate the future situations in the temporary absence of situational data. This ensures that the system is somehow fault tolerant with respect to communication failures. If the complexity and uncertainty of the knowledge is high, probabilistic or fuzzy methods may need to be incorporated. AISim communicates with the simulation system using standard data protocols. Communication with other computer generated or manned simulation entities in the scenario will be in the same way. The exchange of AI player-specific data between AISim and ITEMS is performed using a signal pdu (protocol data unit) which contains application specific data. AISim activates the AI target by sending a guidance order. ITEMS has to confirm the availability of the target for guidance before the execution of the order. If it is available then AISim takes control of the AI target using its knowledge base which is equipped with a pilot's knowledge. AISim checks the player's status by receiving its entity state pdu from the network which is issued by the interface program attached to simulation system. When AISim needs to communicate with another agent, it has to seek information regarding that entity on the network.

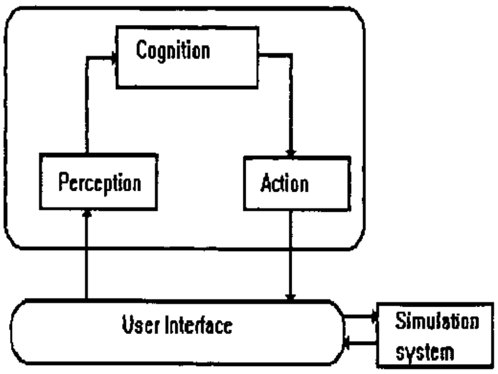

Figure 2: Structural Components of an Intelligent Agent

5. CONFLICT RESOLUTION

It is obvious that knowledge used by an agent can be incomplete and the goals of the agents might be conflicting [6]. When several agents have conflicting goals under uncertain environments, the complexity of the conflict resolution increases. If the agent itself has conflicting goals, a set of heuristics or a classifier can be used to deal with the conflict. However if different agents are having conflicting goals, then there is a need for a negotiator to deal with this problem. The negotiator is an agent which defines the authority of information. The behavior of AISim is controlled by a "task control operator" which defines which task is to be activated. When AISim is connected to DIS network, it will be able to locate its prime opponent among several agents within a scenario and will be able to control the overall interaction when the prime opponent is changed dynamically.

6. LEARNING AND EXPLANATION

Computer generated agents need to be able to learn meaningful actions in various tactical situations and explain the reasons behind such actions. A prototype system called RSIM prior to AISim has been designed to measure the feasibility of implementing machine learning techniques in general, and explanation based learning (EBL) techniques in particular [1]. The results seem to be encouraging, and AISim is expected to be equipped with this capability in the near future.

7. CONCLUSION

Intelligent agents will continue to be one of the active research areas in real time simulation studies. Capabilities provided by AI techniques to deal with uncertain, incomplete, time dependent, and complex environments will lead researchers to develop more effective agents. Hierarchical system architectures have been used in the design of intelligent agents to implement such capabilities. An intelligent agent control system, AISim, developed for this purpose, is now able to automatically perform air interception tasks in an air warfare scenario including take off, navigate, patrol, tactical air combat maneuvers, air to air refueling, return to base and landing. Knowledge representation, communication, cooperation and interaction between the agents in distributed systems still need to be further studied. The behavior of AISim will continue to be improved in this respect. The system is expected to be connected to a DIS network by the middle of 1996.

References

1. Kocabas S., Oztemel E., Uludag M., Koc N., "Automated agents that learn and explain their own actions: A progress report", Proceedings of the 5th Conference on Computer Generated Forces and Behavioral Representations, Orlando, Florida, 1995, 63-68

2. P. Landweer, "Action/cognition behavior model for computer generated forces", Proceedings of the 4th Conference on Computer Generated Forces and Behavioral Representations, Orlando, Florida, 1994, pp. 335-345

3. Oztemel E. and Kocabas S., "Artificial intelligent threats in distributed interactive simulation", Proceedings of the 1st Symposium on System Engineering and Military Applications, Ankara, October 1995, pp. 476-487 (in Turkish)

4. P. S. Rosenbloom, W. L. Johnson, R.M. Jones, F. Koss, J. E. Laird, J.F. Lehman, R. Rubinoff, K.B. Schwamb, M. Tambe "Intelligent automated agents for tactical air simulation: A progress report", Proceedings of the 4th Conference on Computer Generated Forces and Behavioral Representations, Orlando, Florida, 1994, pp. 69-78

5. M. Tambe, W. L. Johnson, R. M. Jones, F. Koss, J. E. Laird, P.S. Rosenbloom, and K. Schwamb " Intelligent agents for interactive simulation environments", AI Magazine, Spring 1995, 15-39

6. R.M. Jones, J. E. Laird, M. Tambe, P. S. Rosenbloom, "Generating behavior in response to interacting goals", Proceedings of the 4th Conference on Computer Generated Forces and Behavioral Representations, Orlando, Florida, 1994, pp. 317-324

7. Technical Report, "Standard for distributed interactive Simulation- Application protocols", Ver. 2.0, 4th Draft, IST, University of Central Florida, IST-CR-93-40, 1994

8. R.D. Beer, "A dynamical systems perspective on agent environment interaction", Artificial Intelligence, 72, 1995, pp.173-215

9. Technical Report, "DIS standard Development, Operational Concepts 2.3", IST, University of Central Florida, IST-CR-93-25, 1993

10. B. Hayes-Roth, "Architectural foundations for real time performance in intelligent agents", In: Second Generation Expert Systems, J. -M. David, J. -P. Krivene, R. Simmons (Eds),Springer Verlag, 1993, pp 643-673

11. M. Adler, E. Durfee, M. Huhns, W. Punch, E. Simoudis, "AAAI workshop on cooperation among heterogeneous intelligent agents", AI Magazine, Summer 1992, 39-42