DIVA: Distributed Intelligent Virtual Agents

S. Vosinakis, G. Anastassakis, T. Panayiotopoulos

Knowledge Engineering Laboratory
Department of Informatics, University of Piraeus
80 Karaoli & Dimitriou str.
Piraeus 18534, Greece
E-mail : themisp@unipi.gr

Abstract. In this paper we present DIVA, an architecture for Intelligent Virtual Environments. In DIVA, systems consist of three different modules. There is a World Server that represents the environment in an abstract way, an arbitrary number of Intelligent Agent Clients acting as the world's inhabitants and a set of Visualisation Clients that generate the system's visual output. All client modules are autonomous and can be dynamically connected and disconnected to the World Server.

1 Introduction

During the past few years, research in the separate fields of Distributed Artificial Intelligence and Virtual Environments has resulted in a significant number of architectures and applications combining elements from both areas, leading to the evolution of the promising field of Intelligent Virtual Environments (IVEs) [1,2,6,9].

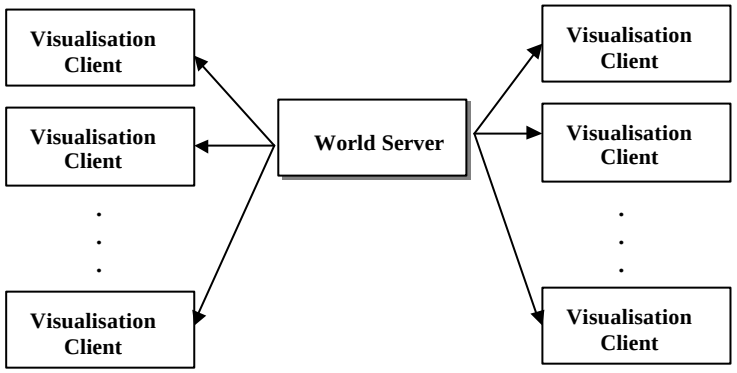
Agent-based technologies have been rapidly emerging since the beginning of the 1990s [3-5]. Agents can be defined as hardware or software systems with adaptive behaviour, acting effectively in dynamically changing environments. Intelligent agents' technology has presented many applications so far, starting from smart e-mail clients to software for robotics applications.

Many efforts have been made towards the introduction of Intelligent Agents in Virtual Environments. One of those belongs to J. Coble and K. Harbinson, who propose a Multi – Agent architecture for Virtual Environments (MAVE) [6]. The virtual environment provides a framework in which distributed users can collaborate and share various resources. Additional efforts have been made in the field of Intelligent Virtual Environments, such as the combination of Artificial Intelligence / Artificial Life and Virtual Environments.

E. Denti, A. Natali and A. Omicini [7] propose a way to merge Logic Programming into Web–based technology. The availability of logic programs on the Internet is also the subject of research done by S. Beltagy, M. Rafea and A. Rafea [8].

What we propose in this paper is an architecture for implementing systems that integrate Logic Programming, Object – Oriented programming and Virtual Reality. This architecture supports the participation of many intelligent agents in a shared world as well as the simultaneous visualisation of their actions in an arbitrary number of clients. Beyond that, it also allows the direct involvement of the user in the virtual environment, and can be easily extended to support the implementation of complex Intelligent Virtual Environments.

2 Architecture



DIVA-based systems consist of three different modules: First of all there is a World Server representing the environment. It does not contain any visual data, but it has an abstract representation of the world and its objects and generates the "senses" of the intelligent agents. It receives any actions that the agents perform, updates its own world representation and generates the appropriate knowledge that the agents should perceive according to their position and state in the virtual environment. Whenever a change in the state of the world or its objects takes place, it transmits a set of commands to the visualisation modules so that the new state is described in a visual way in real time.

A second module is the Intelligent Agent Client, which is an autonomous module that receives knowledge using a certain protocol and generates actions. It has a declarative way of representing knowledge and, having its own goals and rules, it performs planning and reasoning and generates the set of actions that will be applied to the world or its objects. It can be connected to any world server that uses the same protocol to communicate with the agent clients and supports its actions.

The last module is the Visualisation Client, which displays the state of the world in real time using 2D or 3D data. It receives a set of commands from the world server and modifies the virtual objects in real time. Each command refers to one or more agents/objects and their actions/changes of state that took place. If the implemented application requires synchronised visual output, the client should also return an indication that the visualisation process is over.

These three modules are connected via TCP/IP. There is only one server and an arbitrary number of clients, both intelligent agents and visualisation ones. These clients can be connected or disconnected dynamically from local or remote machines while the server is running, and all they need is to support the same protocols that the server is using. When an application requires the additional presence of real users acting as agents, one could also define a third kind of client, which will be introduced as both intelligent agent and visualisation client, supporting also both communication protocols.

2 Implementation

We have implemented an example multi-agent system based on the DIVA architecture. The system is a simulation of a virtual supermarket, where actors have diverse roles such as customers, managers, thieves and security guards. The supermarket is implemented as a DIVA world server and the actors as DIVA agents.

The DIVA world server is implemented as a standalone server application that allows agent and visualisation clients to be connected. It has a general representation of the virtual world and a set of "laws" that determine: a) who can perform which action to whom and b) the circumstances under which a specific action is applicable.

There are two different types of threads that handle the respective client types sharing the same world representation and being able to read or alter its data. When an agent communicates with the world and requests an action, the world server checks its set of laws to determine whether this action will be successfully performed or not. In the first case, it updates the world data as necessary and notifies all visual agents that a change in the world took place. The results are sent back to the agent client using the agent-world communication protocol.

DIVA agents are based on the BDI (Beliefs - Desires - Intentions) architecture and comprise of two modules, a logical core performing all reasoning and knowledge base handling and an implementation module which manages the logical core and handles data interchange between the agent and a DIVA world server. The overall agent operation can be thought of as a sequence of Sense-Decide-Act cycles.

During the sensing process, the implementation module retrieves all available sensory input from the world server, produces its predicate-based symbolic representation and updates the knowledge base using appropriate logical core functions. The purpose of the decision process is to derive the agent's next action. During the process, goal priorities are revised according to the current world state. After that, the first remaining action of the current plan is selected as the next one. Otherwise, if a plan is not available for the current goal, the logical core's planning functions are used to produce an action plan. The action derived by the decision process is first applied on the world by sending appropriate commands to the world server. If application succeeds, the agent-specific effects are processed. Otherwise, specific on-failure assumptions are adopted.

Visualisation clients consist of two modules: a 3D engine that displays the virtual world and an implementation module that communicates with the world server and updates the world's visual attributes as necessary.

3 Conclusions

DIVA-based systems incorporate key disciplines such as agent intelligence, dynamic environment behaviour, virtual world sharing and enhanced three-dimensional representation, providing a powerful mechanism for simulating and visualising real-world situations where intelligent actors are involved. Moreover, the structure of such systems is open to implementation options allowing for incorporation of modern technologies and tools. Nevertheless, there is a further need to define a set of basic actions to standardise the agent-world communication protocol, where more complex actions will be declared as a sequence of the basic ones.

In the future we are planning to develop complex IVEs using the DIVA architecture, such as a driving simulation, an intelligent virtual laboratory [2], etc. Having already developed a language for the description of virtual agents [10], we are also trying to extend its capabilities to support the DIVA architecture and become a tool to describe and generate DIVA-based IVEs.

References

1. Panayiotopoulos, T., Zacharis, N., Vosinakis, S.: Intelligent Guidance in a Virtual University. IMACS/IFAC International Symposium on Soft Computing in Engineering Applications, SOFTCOM '98 (1998)
2. Panayiotopoulos, T., Vosinakis, S., Katsirelos, G., Avradinis, N.: Interactive experimentation in Virtual Laboratories. CAL99 Int. Conference, Virtuality in Education (1999)
3. Muller, J.: The Design of Intelligent Agents, A Layered Approach. Lecture Notes in Artificial Intelligence, Vol. 1177. Springer-Verlag (1996)
4. Wooldridge, M., Jennings, N.: Intelligent agents: Theory and Practice. Knowledge Engineering Review, Vol. 10(2). (1995) 115-152
5. Cavedon, L., Rao, A., Wobcke, W., editors: Intelligent Agent Systems, Theoretical and Practical Issues. Lecture Notes in Artificial Intelligence, Vol 1209. Springer-Verlag (1997)
6. Coble, J., Harbison, K.: MAVE : A Multi – agent Architecture for Virtual Environments. 11th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems.
7. Denti, E., Natali, A., Omicini., A.: Merging Logic Programming into Web-based technology : a Coordination-based approach. Proceedings of 2nd International Workshop on Logic Programming Tools for Internet Applications (1997)

8. El-Beltagy, S.R., Rafea, M., Rafea, A.: Practical Development of Internet Prolog Applications using a Java Front End. Proceedings of 2nd International Workshop on Logic Programming Tools for Internet Applications (1997)
9. Panayiotopoulos, T., Katsirelos, G., Vosinakis, S., Kousidou, S.: An Intelligent Agent framework in VRML Worlds. Third European Robotics, Intelligent Systems & Control Conference, EURISCON '98 (1998)
10. Panayiotopoulos, T., Anastassakis, G.: Towards a Virtual Reality Intelligent Agent Language. 7th Hellenic Conference on Informatics (1999)