# Visualising Logic Programs in Virtual Worlds


T. Panayiotopoulos
University of Piraeus,
Dept. of Informatics
80 Karaoli & Dimitriou str.
18534 Piraeus, Greece
(+301) 4142038
themisp@unipi.gr

S. Vosinakis
University of Piraeus,
Dept. of Informatics
80 Karaoli & Dimitriou str.
18534 Piraeus, Greece
(+301) 4142038
spyrosv@unipi.gr

S. Kousidou
University of Piraeus,
Dept. of Informatics
80 Karaoli & Dimitriou str.
18534 Piraeus, Greece
(+301) 4142038
nova@eexi.gr

L. Balafa
University of Piraeus,
Dept. of Informatics
80 Karaoli & Dimitriou str.
18534 Piraeus, Greece
(+301) 4142038
kalimera@otenet.gr



## ABSTRACT

Logic Programming as well as Virtual Reality applications have achieved a lot in the last few years. In this paper we present an architecture, a methodology and two simple examples for visualising the output of Logic Programs and integrating different technologies such as Logic Programming, Object – Oriented Programming and Virtual Reality. We are currently working on developing a generic tool for 3D visualisation of Logic Programs.



## Keywords

Logic Programming, Virtual Reality


## 1. INTRODUCTION

Logic Programming (LP) has always been a field of continuous experimentation. Recently many attempts have been made to putting together technologies like Logic Programming, Object - Oriented Programming, as well as Virtual Reality. Therefore, a need for a step further was emerging gradually and the last few years it became rather imperative: The need to integrate all these technologies, i.e. to enhance Virtual Reality applications using both logic and object – oriented implants. Quite recently, attempts have been made to integrate Virtual Reality (VR), Desktop Virtual Reality (Desktop VR) in particular, with educational software to produce a higher level of interaction and visualisation. By the term 'visualisation of Logic Programs', we imply the visual representation of Logic Programs. In other words, the perception of the output of a LP in a visual way. Who would not prefer to perceive an intelligent agent walking into a maze rather than reading the printout of each movement action?

There have been done many efforts towards the visualisation of LPs. One of those belongs to Jeffrey Coble and Karan Harbinson, who propose a Multi – Agent architecture for Virtual Environments (MAVE) [17], which an effective platform for groupware and coordination information systems.

What we propose in this paper, is an architecture of implementing systems that integrate Logic Programming, Object – Oriented programming and Virtual Reality. We also present a methodology that clarifies the levels of abstraction that a programmer should bear in mind, in order to integrate all three entirely different technologies. Finally, we apply the proposed architecture and methodology to a simple example, such as Hanoi towers.

## 2. A 3-MODULE ARCHITECTURE

### 2.1 Conceptual Level of Analysis

The architecture we present at the current section consists of two main units: The Logic Unit (LU) and the Visual Representation Unit (VRU). The connection between them is achieved with the use of an auxiliary Unit Interface. Figure 1, shows the connection of all three of them. All three units are abstract concepts and can be divided further.

LU consists of a Solver, which is the engine that produces the result of a problem. LU also contains a Knowledge Base (KB). Its purpose is to store information about the state of the world, spatial knowledge and generally, every kind of information that the Solver would need in order to extract the desired action or sequence of actions. We are interested in LPs, which refer to worlds that can be visualised. Therefore, a part of the LP must refer to the domain knowledge, i.e. the world, while another part refers to actions to be performed (strategic knowledge). Thus, KB can be divided further into Domain KB as well as Strategic KB.

Respectively, VRU contains a Virtual Reality Management Unit (VRMU), which manipulates the visual representations of the objects that take part in a certain problem. VRMU can be a set of instructions that determine the position as well as other features of the objects. VRMU exchanges information with a 3D-Models database, which stores representations of the objects along with their features. VRMU has also an interactive relationship with a Virtual world, where it places the objects. The 3D-Models DB contains all the 3D objects that will be visualised. Those models are divided into static and dynamic. Static are called the ones that are not affected by any possible action during the whole process (e.g. a wall), while the dynamic ones can change their geometry and/or material attributes in real time (e.g. a human).

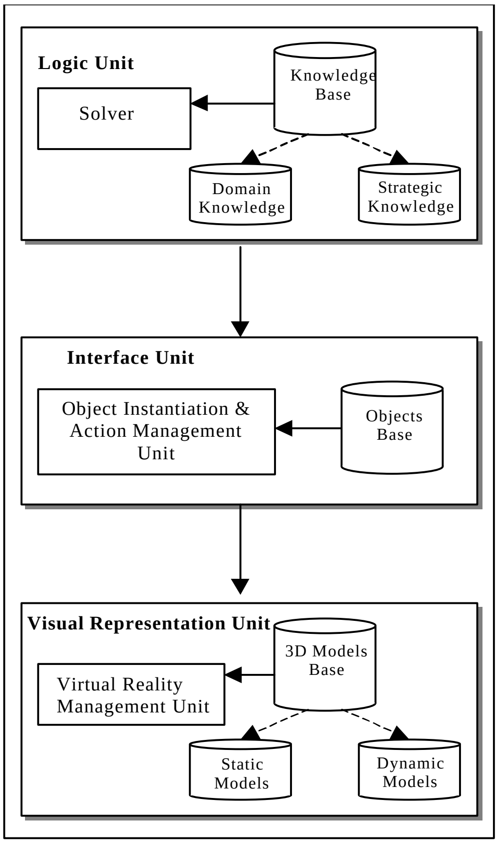


**Figure 1: Architecture**

## 3. A METHODOLOGY FOR LOGIC PROGRAM VISUALISATION IN VIRTUAL WORLDS

In Table 1 we summarise the three levels of abstraction and their responsible parts.

| Conceptual | Instantiation | Visualisation |
|---|---|---|
| Logical constant | Object | 3D model |
| State | Attribute | Geometry / material feature |
| Action | Method | Animation |

**Table 1: Conceptual, Instantiation and Visualisation level**

### 3.1 Towers of Hanoi

The first example is the known problem of the Towers of Hanoi.

To solve the problem, one has to call 'hanoi (N, A, B, C, L)', where:

- N is the number of disks,
- A, B and C are the names of the three pegs and
- L is the output list with the sequence of actions that solve the problem.

Supposing that our pegs are p1, p2 and p3, and the number of disks N = 2, the solution would be L=[p1/p2, p1/p3, p2/p3].

When the user defines the number of disks, the program initialises the appropriate disk objects and displays them on the screen. It then visualises the solution of the problem using the 'moveTo' method to change their position from a peg to another.

## 4. CONCLUSIONS AND FUTURE WORK

In the current section, we will present some possible extensions of what we have seen so far. First of all, we could modify the implementation of such applications in order to include the possibility of running them on multiple machines. A possible scenario would be distributing a problem to a set of clients, who in turn will return the proper sequence of actions to the server.

Secondly, we could provide our current work means of bearing in mind temporal data, in addition to spatial ones. That is, if our sequence of actions is produced by planning techniques, we could take advantage of temporal planning advances [4, 5].

Since our architecture and methodology is proposed for visualising actions, we could apply those to intelligent Agent architectures [6,7].

As a conclusion, we believe that our proposed methodology and implementation provides a simple yet efficient way of introducing logic programming into desktop VR worlds, taking advantage of the most recent Internet technologies (Java and VRML).

## 5. REFERENCES

[1] Marinagi, C.C., Panayiotopoulos, T., Vouros, G.A. Spyropoulos, C..D., Advisor : A knowledge-based planning system. International Journal of Expert Systems, Research and Applications, Vol.9, No.3, pp.319-355, 1996.

[2] Panayiotopoulos, T., Zacharis, N., Vosinakis, S., Intelligent Guidance in a Virtual University. IMACS International Symposium on Soft Computing in Engineering Applications, SOFTCOM'98, Athens, June 1998.

[3] Panayiotopoulos, T., Vosinakis, S., Katsirelos, G., Avradinis, N., Interactive experimentation in Virtual Laboratories. CAL99 International Conference, Virtuality in Education, London, UK, March 1999.

[4] Dede C., Salzman, M., Loftin, B., Ash, K., Using virtual reality technology to convey abstract scientific concepts. In M. Jacobson & R. Kozma (Eds.), Learning the sciences of the 21st century: Research, design and implementation of advanced technological learning environments. Lawrence Erlbaum, Hillsdale, NJ, 1999 (in print).

[5] Panayiotopoulos, T., Katsirelos, G., Vosinakis, S., Kousidou, S., An Intelligent Agent Framework in VRML worlds. Third European Robotics, Intelligent Systems & Control Conference, EURISCON'98, Athens, June 1998.

[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] Panayiotopoulos, T., Anastassakis, G., Towards a Virtual Reality Intelligent Agent Language. 7th Hellenic Conference on Informatics, 1999, Ioannina.