An Architecture to Guide Crowds Using a Rule-Based Behavior System

Elsa Schweiss, Soraia Raupp Musse, Fabien Garat and Daniel Thalmann
Computer Graphics Lab. Swiss Federal Institute of Technology
EPFL, DI-LIG, CH 1015 Lausanne, Switzerland
+41 21 6935248

ABSTRACT

This paper describes a Client/Server architecture to combine the control of human agents performing "intelligent actions" (guided by a Rule-Based Behavior System – RBBS) with the management of autonomous crowds which perform pre-programmed actions. Our main goal being ability to model crowds formed by a large number of agents (e.g. 1000), we have used pre-programmed actions and basic behaviors. In addition, RBBS provides the user with an interface for real-time behavior control of some groups of the crowd. This paper presents how the Server application to control virtual human agent's behaviors using a rule-based system.

Keywords

Multi-agent co-ordination and collaboration, agent architectures, network agents, real-time performance, synthetic agents, rule-based system, human crowds' model.

1. INTRODUCTION

Virtual humans grouped together to form crowds populating virtual worlds allow a more intuitive feeling of presence. However, the crowd is not only needed to create an atmosphere but also should simulate intelligent actions of the group or individuals. We have developed RBBS, a rule-based system for human agents' behavior management, and applied its behavior control to virtual crowds. A Client/Server system has been implemented in order to integrate RBBS and our system to model and control groups of human agents.

2. THE RULE-BASED BEHAVIOR SYSTEM

Non-programmer users can describe some human characters through rules (see Figure 1) and interact with the virtual world during the simulation, by sending some daily-life stimuli in real-time that will induce behavioral actions from human agents group.

Rule1: if ((Laura needs to take the train) and (Laura has no ticket))
then (Laura goes to counter to buy one)
daily-lifeEvent1: at 15 Laura has to take the train.

Figure 1: Example of Rules and Events defined by user

When a daily-life event happens, the set of behavioral rules is checked. If matching conditions are found, then associated conclusions are activated.

As a result, high-level orders are generated. In case of unfamiliar situations, the user is warned that a default response will be induced if no new rules are given. Indeed the user can enter new behavioral information during the simulation.

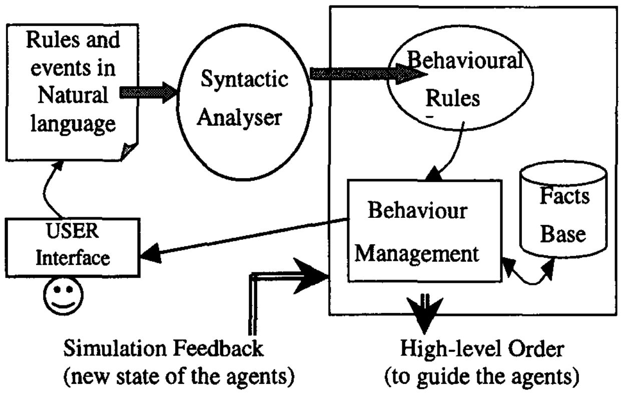


Figure 2: RBBS Organization

Figure 2 shows each step of the RBBS process:

1. The user defines some behavioral rules in a simple syntax close to the English language (see Figure 1)
2. The Syntactic Analyzer translates these definitions into a code understood by the Behavior Management module
3. During the simulation:

   3.1. if no rule is matched then if the user gives a new one, the steps 1 and 2 are done

   3.2. if behavioral rules are selected, high-level orders (like *"go to the counter and buy a train ticket"*) are then generated according to the Facts Base that contains the daily-life events (like *"Actor1 needs to take the train"*) and the current system state (e.g. *"Actor1 has no ticket"*). State changes in succession are induced [5].
4. RBBS is synchronized with the Crowd application by waiting adequate order-feedback.

RBBS is integrated with the Crowd Model (see Section 3) in Client/Server architecture as the behavioral client. It guides the crowd by sending high-level orders to the server according to a fixed communication protocol. Consequently, the RBBS client has to code these orders and decode crowd simulation feedback.

3. THE CROWD MODEL

To simulate a crowd composed of agents endowed with different levels of autonomy, we considered two types of virtual agent behavior control. The first one uses the pre-programmed behavior causing autonomous actions and the second one employs the RBBS guiding of actions (Section 2). "Autonomous groups" are defined to consist of agents whose actions and behaviors are specified before the simulation starts. We have

Autonomous Agents '99 Seattle WA USA

created a script language where the pre-programmed actions can be specified, as shown in the script in Figure 3:

```
GROUP    gr_1
         NB_AGENTS 15
         GROUP_NATURE AUTONOMOUS
         PATH_DEFINITION paths_file
         BASIC_POSTURE postures_file
```

Figure 3: An example of script language to specify groups information

The group called *gr_1* is formed by 15 agents and must follow the scripted pre-programmed information (GROUP_NATURE AUTONOMOUS). The data included in *paths_file* and *postures_file* determine the possible paths (crowd motion) to be followed and the possible initial postures to be applied respectively. Specific actions which must occur during the simulation (e.g. play a keyframe sequence) are associated with the crowd paths, being part of a list of actions to be executed in the time [4].

Apart from the autonomous groups, which follow the pre-programmed behaviors, "guided groups" can be controlled during the simulation. In order to apply this control, we implemented a Client/Server architecture, where RBBS and Crowd clients can communicate. According to a protocol we have devised, the server is able to translate and redirect messages. More details about this process will be presented in Section 4. In the guided control, the Crowd client handles the information about IP (interest points) and AP (action points) that can be specified during the simulation. For example consider an order sent by RBBS: *"go to the tourist office to buy an opera ticket"*. This action can be divided into two parts: a motion command (IP) and an action command (AP). Before sending these data to the Crowd Client, they must be translated by the server in order to be compatible with the information awaited by the receptor.

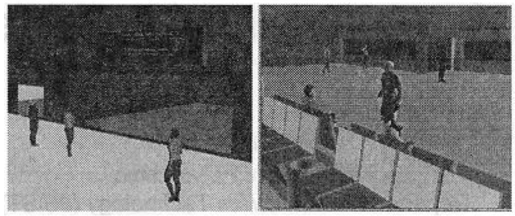
4. THE CLIENT/SERVER

In the Client/Server approach [1][3], the server translates and distributes information from one client to another. We choose to use one socket stream for each guided group. This choice allows having different behavioral clients running simultaneously on different machines.

While RBBS sends high-level orders (a typical RBBS message is *"group 1 must go to shopping; buy one hotdog"*), the message awaited by the Crowd client is at a lower level, like *"Reach position (x, y, z)"*. In order to translate the information between the RBBS and Crowd clients, the server needs a database (DB World) where the environment information is stored (e.g. location of a bus stop, etc). Afterwards, the server is able to translate and transmit the codified order into a command readable by the Crowd client.

5. RESULTS

We have simulated a human crowd management in a train station (Figure 6) in order to mix different levels of control presented in this paper: scripted and rule-based behaviors. Some specific actions were programmed in RBBS (Figures 4 and 5), e.g. *"buy a ticket in the ticket office"*, *"call someone by phone"*, *"check the timetable"*, *"go sit down to wait for someone"* etc. Such actions were applied depending on the events occurred in the simulation (Section 2).



Figures 4, 5: Guided groups checking the timetable and

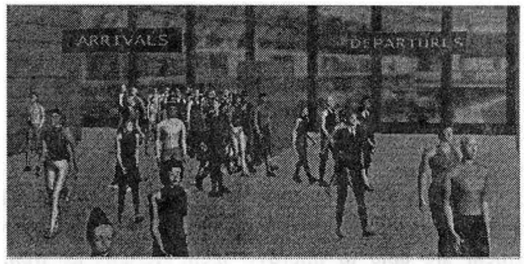


Figure 6: Autonomous groups entering in the train station

The train station simulation involved 80 agents of whom, 16 were RBBS guided.

6. CONCLUSIONS

In this paper we have addressed the control of guided crowd groups using a rule-based system in order to enhance the intelligence ability existing in the virtual agents. In addition, the real-time control feature of our system allows non-programmer users to change crowd behaviors during the simulation. We have evaluated our model in a train station simulation. This project is part of a virtual city project that aims at simulating a virtual urban life [2]. We believe that by increasing the levels of autonomy within the crowd, we are able to obtain more realistic simulation.

7. ACKNOWLEDGMENTS

The research was sponsored by the Swiss National Research Foundation, the Federal Office for Education and Science (ESPRIT eRENA and ACTS COVEN projects), FUNDEPE and CAPES - Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Brazilian office of Education and Science).

8. REFERENCES

[1] Cohen, P.R.; Cheyer, A.J.; Wang, M.; Baeg, S.C. "An open agent architecture", in AAAI Spring Symposium, pp.1—8, March 1994.

[2] Farenc, N.; R.Musse, S.; Schweiss, E.; Kallmann, M.; Aune, O.; Boulic, R.; Thalmann, D. "One Step towards Virtual Human Management for Urban Environment Simulation". ECAI'98, Brighton, England, 1998.

[3] Guzzoni, D.; Cheyer, A.; Julia, Luc and Konolige, Kurt. "Many Robots Make Short Work". SRI International / 1996 AAAI Robot Contest.

[4] Musse, S. R.; Babski, C.; Capin, T.; Thalmann, D. "Crowd Modelling in Collaborative Virtual Environments". Proc. of ACM – VRST'98. November 1998. Taipei, Taiwan.

[5] Norvig, P. "Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp", 1992, M. Kaufmann, ISBN 1-55860-191-0, 946 pages.