Using the BDI Architecture to Produce Autonomous Characters in Virtual Worlds

Jorge A. Torres[1], Luciana P. Nedel[1], and Rafael H. Bordini[2]

[1]Computer Science Institute, Federal University of Rio Grande do Sul, Brazil
{jorget,nedel}@inf.ufrgs.br
[2]Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, U.K.
R.Bordini@csc.liv.ac.uk

Abstract. In this paper, we propose the use of the Belief-Desire-Intention (BDI) model for cognitive agents for the implementation of animated characters. The BDI agent architecture has been widely used in dynamic and complex scenarios where agents may need to act under incomplete and incorrect information about other agents and the environment. In this work, we bring together an articulated model for character animation and an interpreter for AgentSpeak(L), an agent-oriented programming language that implements the BDI architecture. We have developed an interface that allows the BDI-based agent reasoning system to be used for guiding the behaviour of articulated characters in a virtual environment. This is a promising approach for the high-level specification of complex computer animations. The paper also presents a simple 3D animation that illustrates the use of BDI specifications in our approach.

1 Introduction

Autonomous characters in advanced computer games should be able to interact with players and other characters, reacting appropriately to unexpected events and circumstances, and even changing the progress of the game with autonomous courses of actions. The BDI architecture is being widely used in dynamic and complex scenarios where agents may need to act under incomplete and incorrect information about other agents and the environment where they are situated. AgentSpeak(L) [1] is an agent-oriented programming language based on the BDI theory, with some restrictions that are necessary for a practical implementation of those concepts [2].

Various approaches to the development of autonomous animated characters appear in the literature, although most of them do not consider cognitive agents. Perlin and Goldberg [3] developed a system to create believable characters that respond to users and to each other in real-time. Badler et al. [4] developed the PAR architecture to provide a testbed for real-time conversational agents that are situated, communicate, and act on a synthetic 3D world. Funge et al. [5] proposed a cognitive animation model that goes beyond the behavioural model of computer animations, allowing users to control characters' knowledge, and the way it is acquired and used for planning. An important application area of autonomous agents and multi-agent systems deals with the implementation of believable virtual characters, addressing cognitive processes such as emotion, personality and social attitudes [6; 7; 8].

In this paper, we focus on combining high-level specifications of intelligent characters written in AgentSpeak(L) with a framework that assists the generation and animation of 3D articulated figures. We also present a simple case study aimed at giving a short clear example of our approach, rather than demonstrating the potential of the BDI model for the design of complex characters in computer animations.

2 Mind-Body Interface

This section describes an interface that brings together, in an efficient way, the AgentSpeak(L) interpreter (representing the mind) and our framework responsible for modeling and animating 3D articulated characters [9] (the body). We are aware of the importance of cognitive processes such as emotion and personality for achieving believability, although we have not yet addressed them. However, progress in the area of agent-oriented programming with AgentSpeak(L) towards that direction is expected, thus automatically advancing our approach to autonomous characters as well.

2.1 Motion

In order to model an articulated human body, we use a hierarchical data structure for representing the set of joints that compose the body and a set of functions used to generate movement on those joints. Each joint represents a set of possible movements (i.e., a set of degrees of freedom).

In order to build a human body from a set of joints, it is also necessary to specify the topology of such a set. The solution used here was to adopt a tree data structure which approximates the human joints topology, establishing a hierarchical relation between the joints. The joints themselves do not have a geometric representation. Therefore, other structures are necessary to represent the body volume. Such structures have a position and orientation defined in relation to the joint with which they are associated. Thus, the structures that represent the body volume have their positions and/or orientations changed according to joints' new positions and/or orientations [9].

We implement the motion of our characters in three levels. The lowest level deals with the rotation of individual joints. The intermediate level is responsible for encapsulating composed movements that bring together a set of single joint motions. These movements (to which we refer as *tasks*) are loaded when the system is started up and remain available in memory at runtime, providing a higher abstraction level. During the animation execution, the higher-abstraction motion level combines some of the available tasks so as to produce complex behaviour corresponding to the actions chosen through characters' reasoning, taking into consideration contextual information such as position, orientation, and distance to target.

2.2 Reasoning

Since Rao's proposal of AgentSpeak(L) as an abstract programming language [1], considerable work has been done on various aspects of AgentSpeak(L). Some extensions to AgentSpeak(L) were proposed in [10], where a working interpreter for the extended language was also introduced. Bordini and Moreira [11] have given operational semantics to AgentSpeak(L) which they used in the specification of a framework for carrying out proofs of BDI properties for AgentSpeak(L) agents. Also in that paper, the particular combination of asymmetry thesis principles [12] satisfied by any AgentSpeak(L) agent was shown. Such work is relevant in ensuring the rationality of agents programmed in AgentSpeak(L), and represents a step towards formally grounding a simplified version of BDI logics [12] in terms of computable processes.

In the architecture proposed here, the reasoning of a virtual human is implemented with the use of the extended AgentSpeak(L) interpreter [10]. Each agent in the system is a client process running the interpreter with a set of base beliefs and plans (i.e., a source code). The environment is implemented as a server program and manages the information that can be perceived by the agents. It maps the environment state into a symbolic representation that is transmitted to the agents and received by them as percepts (which are used for belief revision). The agents test whether the events generated during belief revision activate any new plan instance. Only one plan can become intended in a single reasoning cycle, and only one intended plan can execute the next part of its body; this can generate new beliefs, goals, or a request for a basic action execution (in the environment) is sent to the interface. New beliefs and goals may serve as triggering events for plans in following reasoning cycles, while actions are sent back to the server to produce changes on the environment.

2.3 Interface

The interface between the animation engine and the reasoning mechanism is responsible for three main tasks: communication between server and clients; storage and control of environmental properties; and control over each behaviour execution.

The communication task involves five conceptual steps (see Figure 1). First, the interface receives a message with the perception information generated by the environment, and the identification of the agent involved. The next task of the interface is to translate the information to the appropriate symbolic format and, using the identification of the agent, to transfer it to the corresponding reasoning process (i.e., a client running the AgentSpeak(L) interpreter). After that, it receives an action request from the interpreter and decides which movement behaviour to adopt. Next, the interface initiates the execution of that predefined behaviour. The cycle is then restarted by observing the environment state, sending perception, and so forth.

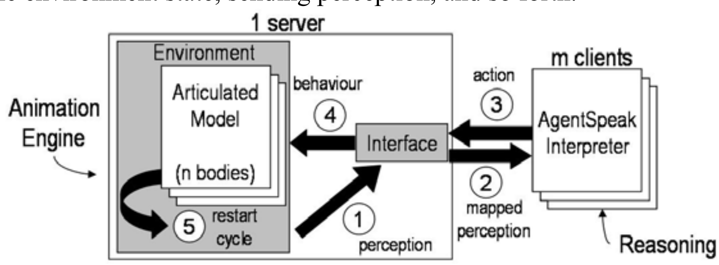

Fig. 1. Architecture of the Animation System: the number of character bodies (n) is only different from the number of agent clients (m) if the application uses avatars

AgentSpeak(L) makes it easy to establish the agents' beliefs, goals, and plans that will guide their reasoning. Information about the environment (e.g., an agent's location) is continuously acquired by perception and belief revision. The interface is responsible for providing perception for each agent; in order to do that, it has structures to store everything that is perceptible by the agents at any time. The interface must then control in which moments, and to which agents, those percepts ought to be sent.

The interface can cope with the various agents being in different stages of a reasoning cycle. For example, while some agents are receiving perception, others can be sending action requests, or having action requests being executed. The interface also needs to keep control over the rendering of the particular movements being performed by each of the agents.

3 Case Study: The Chemical Storehouse

The scenario used to illustrate our approach is that of a robot in a storehouse (see Figure 2). The robot is in charge of storing boxes that are brought into the storehouse from the production line. The boxes can be sensed by the robot when they appear in a place within the storehouse which connects with the production line. The boxes are coloured, and they need to be stored in shelves within the storehouse. However, in a single shelf, only boxes of the same colour can be stored. Also, once a shelf has been used to store boxes of a certain colour, only boxes of that same colour can be stored there, even after the shelf has been emptied. This is relevant, for example, in storing chemical products or in other safety-critical applications.

Fig. 2. The Storehouse Robot

Besides the storage procedure mentioned above, the robot also has to retrieve boxes. Members of staff, which are represented by avatars, can enter the storehouse and request one box of a certain colour. The robot always gives priority to handling these requests rather than storing new boxes arriving from the production line. A box of a particular colour is requested by pressing one of four coloured buttons on the counter at the entrance of the storehouse.

Some animation examples and the AgentSpeak(L) specification used in this application can be found at URL <http://www.inf.ufrgs.br/cg/ras>.

4 Conclusion

We have described briefly the integration of the high-level specification of autonomous characters using AgentSpeak(L) with articulated characters of a computer animation system. With AgentSpeak(L), we can produce characters with elaborate reasoning about acting in an environment. However, the AgentSpeak(L) interpreter assumes that sensorial information will be available to the agent (possibly incomplete, and even incorrect due to sensor failures). Therefore, the interface between the animation system and the agents needs to handle perception of the environment. Our work does not make use of virtual sensors yet. Nevertheless, each agent has access to information from the environment that is partial and dependent on the agent's context.

Although our motivation is similar to Funge et al. [5], as we aim at using cognitive agents, we have focused on the use of AgentSpeak(L), following important developments in the area of agent-oriented programming and multi-agent systems. Computer animation is an area of application where, to the best of our knowledge, the BDI architecture has not been used systematically, yet many types of animations require characters with the type of sophisticated reasoning that AgentSpeak(L) agents display. Thus, our work is a first step towards such use of BDI agents. In future work, we plan to build more complex applications, showing the advantages of the BDI architecture in the implementation of animated embodied agents in virtual environments.

References

1. Rao, A. S. 1996. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. In Proc. MAAMAW-96. Springer Verlag, LNAI 1038, pp. 42–55.
2. Rao, A. S. and Georgeff, M. P. 1995. BDI Agents: From Theory to Practice. In Proc. 1st Int. Conf. on Multi-Agent Systems (ICMAS'95), San Francisco, CA, 312–319.
3. Perlin, K. and Goldberg, A. 1996. IMPROV: A System for Scripting Interactive Actors in Virtual Worlds. In Proc. of SIGGRAPH 96, 205-216.
4. Badler, N. I., Bindiganavale, R., Allbeck, J. M., Schuler, W., Zhao, L. and Palmer, M. 2000. Parameterized Action Representation for Virtual Human Agents. Embodied Conversational Characters, Cambridge, MA, MITPress.
5. Funge, J., Tu, X. and Terzopoulos, D. 1999. Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters. In Proc. of SIGGRAPH 99, Los Angeles, CA.
6. Bates, J., Loyall, A. B. and Reilly, W. S. 1992. An Architecture for Action, Emotion and Social Behavior. Technical Report CMU-CS-92-144, School of Computer Science, Carnegie Mellon University.
7. Rousseau, D. and Hayes-Roth, B. 1997. Improvisational Synthetic Actors with Flexible Personalities. Knowledge Systems Laboratory, Report No. KSL 97-10, Stanford, CA.
8. Marsella, S. and Gratch, J. 2002. A Step Towards Irrationality: Using Emotion to Change Belief. In Proc. 1st Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), Bologna, Italy.
9. Maciel, A., Nedel, L. P. and Freitas, C.M.D.S. 2002. Anatomy Based Joint Models for Virtual Humans Skeletons. In: IEEE COMPUTER ANIMATION, Geneva, Switzerland.
10. Bordini, R. H., Bazzan, A. L. C., Jannone, R. O., Basso, D. M., Vicari, R. M. and Lesser, V. R. 2002. AgentSpeak(XL): Efficient Intention Selection in BDI Agents via Decision-Theoretic Task Scheduling. In Proc. 1st Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), Bologna, Italy.
11. Bordini, R. H. and Moreira Á. F. 2002. Proving the asymmetry thesis principles for a BDI agent-oriented programming language. In Proc. of CLIMA-02, held with FLoC'02, Copenhagen, Denmark. Electronic Notes in Theoretical Computer Science, 70(5).
12. Rao, A. S. and Georgeff, M. P. 1998. Decision procedures for BDI logics. Journal of Logic and Computation, 8(3):293–343.