BDI Model-Based Crowd Simulation

Kenta Cho, Naoki Iketani, Masaaki Kikuchi, Keisuke Nishimura,
Hisashi Hayashi, and Masanori Hattori

TOSHIBA Corporation
Komukai-Toshiba-cho, Saiwai-ku, Kawasaki-shi, 212-8582, Japan

Abstract. We present a new approach for crowd simulation in which a BDI (Belief-Desire-Intention) model is introduced that makes it possible for a character in a simulated environment to work adaptively. Our approach allows the character to realize realistic behavior by adapting its action with the sensed information in a changing environment. We implemented a demo system simulating the BDI model-based NPCs that extinguishes a forest fire with a 3D game engine, Source Engine. We measured the performance to evaluate the scalability and the bottleneck of the system.

1 Introduction

Recently, crowd simulation has attracted attention as a technology that realizes a realistic metaverse, including many intelligent characters, and simulates a social environment on the basis of interactions among characters. Human characters in crowd simulation should act as if they are real humans to realize a realistic simulated environment, but it is not easy to implement such realistic characters that are capable of diverse actions, act adaptively in the environment and cooperate with one another.

We propose crowd simulation that focuses on character adaptability in a simulator by applying a BDI (Belief-Desire-Intention) model to characters. In this paper, we explain our BDI-based approach and an implementation of crowd simulation based on that approach. Section 2 describes why we use the BDI model in crowd simulation and in section 3 and 4 our crowd simulation prototype system is explained. We evaluate that system in section 5 and present our conclusions in Section 6, the final section.

2 Approach

Crowd simulation is an important technology for adding realism to simulated environments such as metaverses and games. For a human character in a simulation to be experienced as realistic, it is necessary to satisfy certain requirements. [Lankoski 07] shows some design patterns to endow an NPC (non-player character) with human qualities so that it is experienced by a player as if it were a real person. Those qualities are related to human body, self-awareness, intention states, self-impelled actions, expression of emotions, ability to use natural language and persistent traits.

H. Prendinger, J. Lester, and M. Ishizuka (Eds.): IVA 2008, LNAI 5208, pp. 364–371, 2008.

Although all these qualities are important for making a realistic NPC, we focused on adaptability of NPCs. Adaptability is related to self-awareness, intention states, self-impelled actions and persistent traits. An NPC that works to solve continuous goals should change its action when it senses a change in the surroundings and act adaptively with respect to the perceived information.

We use the BDI model to realize the adaptability of NPCs. The BDI model selects an NPC action based on 3 states of mind: belief, desire and intention. A belief represents the NPC's knowledge, including information about an external state and an internal state of the NPC. A desire represents what he/she wants to do at that moment. An intention represents what he/she is doing or is deciding to do at that moment. The BDI model is described as suitable for modeling a real-time system [Rao 95]. Since crowd simulation that adapts to a changing situation in real time is a real-time system, we adopted an approach in which the BDI model is used to implement NPCs in crowd simulation.

We implemented a demo system simulating NPCs that extinguish a forest fire. The NPCs work adaptively according to changes in the environment such as the spread of the fire and the changing positions of other NPCs. Since we thought a forest fire is a situation requiring NPCs to act flexibly according to the changing surrounding environment and cooperate with each other to overcome the problem, we use it as a test bed for our BDI-based crowd simulation.

3 BDI Model-Based Crowd Simulation Demo System

3.1 Overview

In this section, we explain our BDI-based crowd simulation demo system and its architecture. Each NPC in the system works according to 3 states of mind, namely, belief, desire and intention, and adapts its action to the change of surroundings such as the positions of fire, water and other NPCs and the spread of fire.

The system has a module called a perceiver that abstracts the information from surroundings, which is gathered through the NPCs' senses of vision and

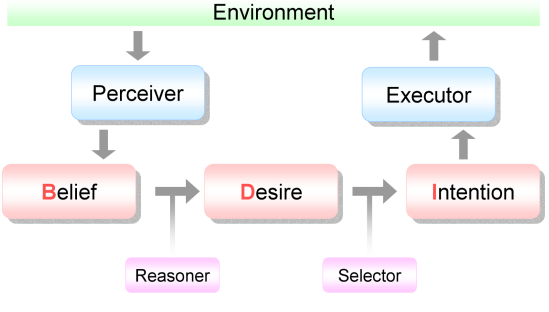

Fig. 1. BDI model

hearing, and derives meanings from that information. These meanings are used as beliefs of NPCs. There is another module called an executor that maps an NPC's intention into the NPC's concrete actions in the environment. These two modules work as an interface between the environment in the simulation and the BDI model (Fig.1) in the NPC.

This system is implemented through the combination of our BDI-based AI system and Source Engine, which is a 3D game engine developed by Valve Software. Source Engine is a well-known platform used for AAA titles such as Half-Life. Source Engine provides many features, such as 3D rendering, character animation and physics simulation, and by using it our demo system can be developed in a short term.

3.2 Scenario

In this section, we show the typical working scenario in the system.

- In the first stage, NPCs don't know that a fire has occurred, and are talking to one another in the village.
- A NPC walks to the forest and discovers a fire.
- The NPC goes to the nearest river and tries to extinguish the fire.
- The fire spreads and the NPC thinks it will be difficult to extinguish all the fires by itself.
- The NPC returns to the village to get the help of other NPCs. (Fig. 2)
- Other NPCs are starting to extinguish the fire separately.
- The fire spreads faster and NPCs try to form a bucket brigade. (Fig. 3)
- Since the fire spreads faster and faster, NPCs think it is impossible to get a handle on the fire and run away.

Fig. 2. Get the help of other NPCs

Fig. 3. Bucket brigade

This is one of the scenarios that can occur in this system. If there are many NPCs or the NPC finds a fire faster, the fire can be extinguished. The result of the simulation changes according to the parameters such as the number of NPCs, location of NPCs and fires, and the speed at which the fire spreads.

4 Architecture

In this section, we explains the BDI model that provides the adaptive actions of NPCs. There are 3 data structures (belief, desire, intention) that represent the inner mind states of the NPC and 4 modules (perceiver, reasoner, selector, executor) that manage and update these structures.

4.1 Perceiver

The perceiver perceives the surroundings and creates and updates beliefs. Information about the surroundings is gathered by Source Engine. Source Engine brings the information of vision and hearing in a limited range from the NPC to the perceiver, and the perceiver updates beliefs. Beliefs that include the information about the environment in the map are represented with the location data of the grid where the information is perceived. The grid is used to represent the abstracted data of location-based perceived information. Each belief has a weight that represents the importance of each event for the NPC. For example, the weight of the belief about the fires in each grid represents the number and the intensity level of fires in the grid.

4.2 Reasoner

The reasoner retrieves desires according to the updated beliefs with the pre defined rule set. A rule fires for a given group of beliefs and derives corresponded

desires. Types of desires include stand talking, take water (from where), pour water (to where), run away, cry out (at where, what) and form a bucket brigade (from where to where). The number of each type of desire is not limited to 1. For instance, take water desires can be generated according to each grid where the NPC can take water.

Each desire has a weight that represents its importance. The rule that retrieves the specific desire calculates the weight according to the weights of the corresponding beliefs. For example, the weight of the pour water desire is calculated based on the distance to the corresponding grid and the number of other NPCs in the grid. The number of NPCs is used for avoiding a congested grid where many NPCs rush to pour water.

4.3 Selector

The selector selects the desire that has the highest weight's value and uses it as the NPC's intention. To ensure that the NPC does not change its intention too frequently, the selector changes the current intention when the weight's value of the current desire is larger than the multiplied weight's value of the current intention.

4.4 Executor

The executor converts the current intention into a concrete sequence of actions and these actions are executed with Source Engine.

4.5 Applying BDI Model to Our Demo System

By using the BDI model, the mechanism to control the NPC can be separated easily into two layers. One layer handles the higher-level estimation for the action of the NPC such as which fire should be extinguished first or whether work should be executed separately or collaboratively. These estimations are handled with the BDI model that can get a current overview of the environment from beliefs and generate appropriate desires according to beliefs. The other layer handles the lower-level estimation such as finding the path to the fire and the water. In our system, the executor performs the role of this layer.

Since the BDI model in the higher layer only has the abstracted surroundings information modeled as beliefs and higher-level action modeled as intentions, the developer can focus on implementing the sophisticated actions of the NPC within the BDI model.

5 Evaluation and Discussion

We evaluated the performance of the system, and compared our BDI-model based approach to other approaches for the crowd simulation.

5.1 Performance Evaluation

We measured the performance in order to evaluate the system's scalability and bottlenecks. Our evaluation environment is described below.

CPU Xeon 3.2GHz
Memory 3.0GB
OS Windows XP SP2
Profiler VTune Performance Analyzer ver.8

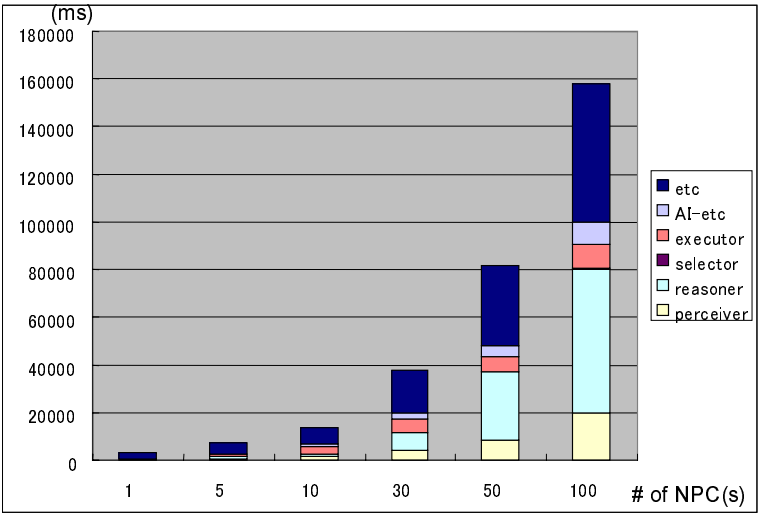

Fig. 4. Performance evaluation

Fig.4 shows the CPU time (micro sec.) of each simulation cycle. The simulation cycle represents the load to control all NPCs in the simulation. Concretely, it is the length of the time in which the GameFrame() method is executed once in Source Engine. Each item in the CPU time means the processing time of the specific process described below.

- Perceiver, Reasoner, Selector, Executor
  Processing time of each component
- AI-etc
  Processing time of the RunAI() method except for the time spent for the 4 components described above. The RunAI() method that controls the behavior of NPCs.
- etc
  Processing time of the GameFrame() method except for the time spent for the RunAI() method, including the process for spreading fires and physics simulation.

The increase in processing time is roughly proportionate to the number of NPCs, and the reasoner's AI-related processes accounted for most of the time. The processing time for the reasoner increased more sharply than for the other items. This sharp increase is attributable to the process in the reasoner to update the beliefs about other NPCs in the simulation and apply rules to these updated beliefs.

The current implementation of the reasoner applies a rule every time when one of the corresponding beliefs for the rule is updated. To minimize the overhead in applying rules to updated beliefs, the reasoner should make a prior evaluation as to whether the change of the belief has a compelling effect on the rule and apply it only when the effect is significant for a retrieved desire.

5.2 Comparison of BDI Model with Other Approaches

Most BDI-model agent systems are implemented based on a reactive planning [Howden 01] [Braubach 05] [Bordini 06], and our BDI model also retrieves the desire reactively with rules. Some systems support communications between agent [Braubach 05] [Bordini 06] [Howden 01] and cooperative action strategy [Bordini 06] [Howden 01]. In our demo system, communication is realized with the voice and cooperative action is realized with the executor that causes NPCs to engage in cooperative action such as formation of a bucket brigade.

Our current approach realizes reactive action by creating desires with the reasoner and continuous action by keeping an intention until the weight of other desires rises above the weight of the current intention. Our approach is advantageous in terms of the simplicity of the system since the BDI model can realize both the adaptabilities and the ability to accomplish a desire, but sequential actions of the NPC cannot be specified. This makes it difficult to set a specific scenario for the NPC since the sequence of the NPC's actions has to be controlled by adjusting the weight of the desire calculated with the rule.

A planning mechanism will be of help regarding this problem. Offline planning [Fikes 71] creates a static plan to achieve a goal. Online planning [Ambros-Ingerson 88] [Wilkin 88] [Hayashi 06] can change the created plan dynamically while the plan is being achieved according to the changing situation. HTN planning [Tate 77] [Wilkin 88] [Nau 99] [Hayashi 06] breaks down an abstract task into concrete tasks in the process of creating a plan. By using the planning mechanism, we can set the specific sequence of actions for the specific desire of the NPC. The planning mechanism can be introduced to our system by mapping an intention to the specific plan.

Another problem in our approach is lack of a cooperating mechanism. We implemented collaborative behavior of NPCs by transmitting beliefs with voices, but it doesn't support the assuming of the leadership of a group of NPCs. A collaborative strategy in multi-agent planning [Ferber 99] will help to solve this problem. For instance, we could introduce a leader agent in our system and the leader agent would create the global plan and assign the corresponding tasks to other agents.

6 Conclusion

We have introduced the approach of the BDI model-based crowd simulation and simulated NPCs that extinguish a fire in a forest. Beliefs of an NPC simulate a limited sense of the surroundings, desires simulate concurrent candidate actions and an intention simulates an adaptively selected action. The approach presented in this paper endows NPCs in a simulated environment with more adaptive and realistic behavior. Currently, our work is directed toward striking a balance between adaptabilities and easy description of NPCs by introducing a planning mechanism to our approach.

References

[Lankoski 07] Lankoski, P., Bjork, S.: Gameplay Design Patterns for Believable Non Player Characters, Situated Play. In: Proceedings of DiGRA 2007 Conference, September 3 (2007)

[Rao 95] Rao, A., Georgeff, M.: BDI Agents: from Theory to Practice. In: ICMAS 1995, pp. 312–319 (1995)

[Howden 01] Howden, N., Ronnquist, R., Hodgson, A., Lucas, A.: JACK: intelligent agents - summary of an agent infrastructure. In: IAMSMAS 2001 (2001)

[Braubach 05] Braubach, L., Pokahr, A., Lamersdorf, W.: Jadex: A BDI-agent system combining middleware and reasoning. In: Software Agent-Based Applications, Platforms and Development Kits, pp. 143–168. Birkhauser Book (2005)

[Bordini 06] Bordini, R., Hubner, J.: Jason: A java-based interpreter for an extended version of AgentSpeak, Manual Version 0.8 (2006)

[Fikes 71] Fikes, R., Hart, P., Nilsson, N.: STRIPS: a new approach to the application of theorem proving to problem solving. Artificial Intelligence 2, 189–208 (1971)

[Ambros-Ingerson 88] Ambros-Ingerson, J., Steel, S.: Integrating planning, execution and monitoring. In: AAAI 1988, pp. 735–740 (1988)

[Wilkin 88] Wilkins, D.: Practical Planning. Morgan Kaufmann, San Francisco (1988)

[Hayashi 06] Hayashi, H., Tokura, S., Hasegawa, T., Ozaki, F.: Dynagent: An incremental forward-chaining HTN planning agent in dynamic domains. In: Baldoni, M., Endriss, U., Omicini, A., Torroni, P. (eds.) DALT 2005. LNCS (LNAI), vol. 3904, pp. 171–187. Springer, Heidelberg (2006)

[Tate 77] Tate, A.: Generating Project Networks. In: IJCAI 1977, pp. 888–893 (1977)

[Nau 99] Nau, D., Cao, Y., Lotem, A., Munoz-Avila, H.: SHOP: simple hierarchical ordered planner. In: IJCAI 1999, pp. 968–975 (1999)

[Ferber 99] Ferber, J.: Multi-agent systems: an introduction to distributed artificial intelligence. Addison-Wesley, Reading (1999)