An Abstraction Framework for Cooperation Among Agents and People in a Virtual World

David McDonald, Alice Leung, William Ferguson, Talib Hussain

BBN Technologies, 10 Moulton Street, Cambridge, MA 02138

{dmcdonald, aleung, wferguson, thussain}@bbn.com

Abstract[·]

With increasing sophistication in modeling interpersonal interactions, including cultural and social factors, it is now possible and desirable to bring together state-of-the-art human behavior models (HBMs) developed through different approaches in a framework that facilitates interoperability, composability, and comparison. Such a framework is needed in order for game or simulation environments to allow heterogeneous agents (synthetic characters driven by HBMs) to interact with each other and to interact with human players, even though their information needs and operating premises may differ. An important part of such an interoperability framework is an agent message system supporting multiple levels of abstraction. To demonstrate the viability of such a framework, we created an environment that supported synthetic characters interacting with each other and a human player in a common scenario. The agents were provided by different researchers built upon different technical approaches, yet were able to communicate through messages featuring multiple levels of abstraction to describe actions and events. In this paper, we describe our choices of message abstraction levels, what we developed, and the architecture of the framework that we used to mediate the interaction among all the participants, synthetic and human.

Overview

Many of today's games provide immersive environments in which synthetic agents (also known as non-player characters (NPCs) or game artificial intelligence (AI)) interact with human players in a variety of ways in order to enrich the gaming experience and provide plot elements. Almost without exception, gaming platforms require that these agents be developed using a uniform mechanism, such as a common, centrally-designed set of messages and/or a single embedded scripting language. We suggest that in the long term these restrictions will limit the richness and variety of agents that can be incorporated in a game. The inevitable requirement for complex agent behaviors will force robust and flexible gaming engines to support agents based on diverse mechanisms, encourage re-use of existing agent controllers, and promote intelligent interactions among agents as well as between agents and human players.

In this paper, we present a framework that we have developed for virtual environments in which multiple agent controller mechanisms are incorporated to control different types of agents within the game in different ways. The resulting system allows agents to interact with each other and with the human players, even though their information needs and operating premises differ. In particular, the framework allows for the use of agent control models that are designed to interact with the simulated world (and with the people and other models) at different *levels of abstraction*.

In this paper, we describe the five levels of abstraction (*perceptual*, *literal*, *semantic*, *interpreted* and *narrative*) that we use in our framework. We then give a high-level view of a generic architecture for enabling heterogeneous agents and people to participate in a common virtual world. We finish by describing a specific game-based multi-agent virtual environment which demonstrates these ideas and by discussing where we are headed in future work.

Relevance to entertainment

Most current games, even the largest of them, are built with a "Tower of Babylon" approach – all of the agents and any other AI controlled entities interact through a uniform, centrally-designed set of messages. This approach avoids many of the problems that this paper tries to solve. A unified approach is good engineering if the agents are simple, uniform, and managed by a small development team. These conditions are mostly true in current games. In typical action games (e.g., first-person-shooters), NPCs tend to be limited in their behaviors and do not initiate the same wide range of actions that are available to the player. Generally, the behaviors are scripted and event-dependent, although some recent games have incorporated more open-ended NPC behaviors. In typical strategy games, the NPC and player roles are more symmetric (the AI "plays the game" against the player), but the world is structured by the nature of the strategic game. (Subplots and rich NPC interactions are not standard features of strategic games or they are kept separate from the strategic portions.)

As agents become more complex and diverse, and as development teams become larger and more distributed,

the gaming world Tower of Babylon may fall (or be pulled down). When agents speak multiple (development) lan guages, game creators will need to build systems that can deal fluidly with this complexity. There are several specific reasons that this complexity may be worth the cost or be unavoidable:

Future games may benefit from multiple kinds of NPC "behavior generators." Different NPCs might be run by different scripting engines because they were created by separate development teams. AIs may have to fill radically different roles such as individuals vs. whole "factions." (This happens in some games now.) Even in the case of AI for individuals, NPCs that play the companions of the player's character might need radically different AI than that needed for driving monsters or bystanders. If these NPC controlling engines need to operate at different levels of abstraction, then they should benefit from the architectural suggestions we make here.

Another utility of multiple levels of abstraction is support of non-programmer (level designers or players) authorship of behavior. Different authors may write agents that require or act on information at different levels. Also some behaviors may be much easier to describe at higher levels of abstraction while others may need to be specified (say for dramatic effect) at very low levels. A representation system and architecture that facilitates easy translation among levels should reduce the burden on script writers.

It could be fun to model misinterpretations of lower level descriptions. Imagine an NPC who could make puns or an incompetent NPC who took all orders in a painfully literal way. Of course, such NPCs could not even exist until games are developed that use natural language, but a wide variety of human behavior requires low-level processing that could be intractable if used by all agents or in all circumstances.

Game creators may want to begin to include high level annotations on the events and messages that pass through their systems. Metadata about the dramatic function of events is sneaking into experimental games now and may make the mainstream soon. Our "narrative" level of abstraction is a convenient, uniform way to store such information.

The Levels of Abstraction

In a typical game, agents need to pass along information about three kinds of behaviors: things that they say, gestures that they make, and actions that they take. Complex agents also need to be able to receive information in order to understand events in the game well enough to determine what response or general activity to make. Both for receiving information about the world and conveying information about their activities, different agents may operate at different levels of abstraction.

For example, we have been working with several different *human behavior models* for controlling agents. Each one has its own models of perception and acting. Some respond to very abstract events like being insulted; some to more concrete events like being physically approached. Some produce concrete action directives like *utter the words "Go away;"* others produce abstract directives like *display anger*.

In the real world, people operate at multiple levels of abstraction constantly, and are very good at analyzing their perceptions to determine what someone else has done and why. However, they may make significant mistakes in understanding if they interpret an action at the wrong level: one culture's sign for success (e.g., thumbs-up) can be another's obscene insult. A person that is unaware of this additional cultural level of information is likely to act inappropriately or draw inappropriate conclusions from what they observe.

In a game where agents only need to interact with human players, there is seldom any need to convey information about the intentions behind the behaviors of the agents – the human players are expected to make their own interpretations. However, in order to enable different agents to interact with each other as well as with humans in a virtual environment, it is important to provide the agents with the capability to understand and behave at different levels of abstraction. In both cases, though, the virtual world simulator needs understandable concrete directives about agent behaviors that rendering clients can turn into animations and sound.

We have developed a five-level language for describing agent behaviors. Each level is a valid description. They differ in how far they have been abstracted from raw images and sound or, conversely, how far removed they are from purposes and intentions. Every action that an agent takes is encoded in terms of this language, to the extent that it can be, and every agent sees the actions of the other agents in these terms.

In this section, we describe each of the levels in turn, from the most concrete to the most abstract. In the section that follows, we show how these levels are mapped to communications between the agents, the game client, and the world-model that mediates between them.

Level 1: Perceptual

At the lowest, raw *perceptual level*, the flow of activity in the simulator is "represented" by the audio and video that a rendering client can produce. Utterances are sound, gestures are sets of pixels or geometric transforms and actions are some combination of the two. At this level, messages contain raw information with little to no annotation. In order to process this information, the AI agent must apply perceptual mechanisms directly. For example, a statement may be provided as an audio clip, and speech recognition would need to be applied to process it.

Few game AI agents currently have perceptual or manipulation mechanisms that would allow them to operate at this level and we doubt that there will be much call for them in typical game domains. However, this level would be important for some purposes, such as constructing robot test-beds, in which direct perception of the world (e.g., via speech processing or pattern

recognition) is a critical capability. Some games provide NPCs that simulate the act of perception (e.g., detecting players that are hidden in shadows). However, the underlying algorithms tend to be based on rules applied to higher-level information about objects and properties of the world, rather than direct observation of the simulated environment.

Level 2: Literal

At the *literal level*, the flow of time and activity in the simulator is broken into events. At this and subsequent levels, these events are annotated with machine-readable, symbolic descriptive information. A given event might have annotations from any or all of levels two through five.

At the literal level, utterances are represented as a single event annotated with the speaker and a list of words and prosody information (if possible), as well as descriptions of the coordinated non-iconic gestures and facial expressions that accompany the speech. Iconic gestures are represented as events, which include a physical description of the motions that occurred and who performed them. Actions are represented in world centric terms, such as absolute coordinates for motion, and other primitives that are natural for the simulator. (In systems where the simulator represents the world at a more abstract level, the literal and semantic annotations for action may be identical.)

Level 3: Semantic

At the *semantic level*, events are annotated with a symbolic representation of their content. We use the term semantic because when the event is an utterance, the annotation at this level resembles the interpretation that a good semantic parser would produce.

Utterances are annotated with their "naive" meaning. So "*I'm cold*" will be represented as a statement about temperature rather than an indirect request to make the speaker warmer, and "*Do you know what time it is?*" would just be represented as a query about a capacity to provide knowledge. Gestures are annotated by an unambiguous (though perhaps vague) representation of their meaning as the agent making the gesture intended it. Actions are annotated in functional, scenario-relevant, terms such as move-to(door-1), rather than the spatial coordinates that appear at the literal level.

Level 4: Interpreted

Interpreted annotations are the richest of the levels in our framework. Interpretations include the intent of the performer of the event. They may also include suggested responses or intended consequences. Meanings of or responses to events can also be provided by other components (besides the instigator of the event). Information at the interpreted level may be self-contradictory (and often will be if provided by different agents). The social model (see section 4) may also annotate events at the interpreted level by providing cultural or context specific interpretations of events (or possible responses, etc.).

Level 5: Narrative

At the highest level, the purpose, role or function of events is included in the annotations. This purpose comes from something external to the simulated world, even external to the representations held by the agents in that world. Narrative annotations on events can indicate their dramatic function in a story (e.g., foreshadowing, building suspense, surprising the player, misleading him or her etc.). If the virtual world is serving some purpose other than to entertain, then the narrative annotations can relate to that purpose as well. A simulation that is meant to teach might annotate events as examples of pedagogically relevant phenomena that are to be observed, evidence, or counter-examples.

The components that infer or rely on narrative annotations are not the participants in the world, but rather the shapers, measurers, authors and other entities whose purpose is to ensure that the whole system (including people, agents, simulators, renders, etc.) carries out its purpose (entertains, teaches, extrapolates outcomes, etc.)

While human participants may well be aware of or infer the narrative purpose for events in the simulated world, they often differentiate between what is happening inside the simulation and what its meaning might be in the larger world. A student may realize that her synthetic companion has been wounded in order to giver her a chance to practice first aid skills, but that is a different sort of knowledge than the nature of the virtual injury, which is part of the simulated world.

Many interesting questions revolve around the effect narrative knowledge has on the participants in a virtual world. Do they learn more or less because they know that the world is fictional? How does such knowledge interact with the much-vaunted notion of "immersion"? Many media theorists have written extensively about this dual awareness on the part of participants in a virtual space, but we are unaware of any empirical work that would guide a designer as to the best mix of fiction awareness and complete immersion for various purposes.

Coupled-Worlds Architecture

To support communication among agents and between the agents and the interface to the human players, we developed the coupled-worlds architecture shown in Figure 1. The labels on the lines that connect the components show which levels are passed between them. Double-headed arrows mean that information on that level is passed both ways. We've used solid dark grey for components and links that one would expect in a "conventional" game, and dotted light grey for what we have added.

On the top left, we have the game client. This is where human users access the scenario. The solid arrow connecting it to a person is double headed because it will

carry events initiated by the player's GUI interactions back to the virtual world. The line linking it to the synthetic characters is to remind us of the possibility that some AI agents might well want the raw data and we should not discount that option.

On the right is a pair of tightly-coupled blackboards that provide shared world models for the benefit of the agents. We distinguish physical from social models first to emphasize that the semantic and interpreted levels convey interpersonal information that will not make sense outside of the cultural and social situation playing out in the scenario, and also to reflect the fact that additional work is being done by the combination of the two.

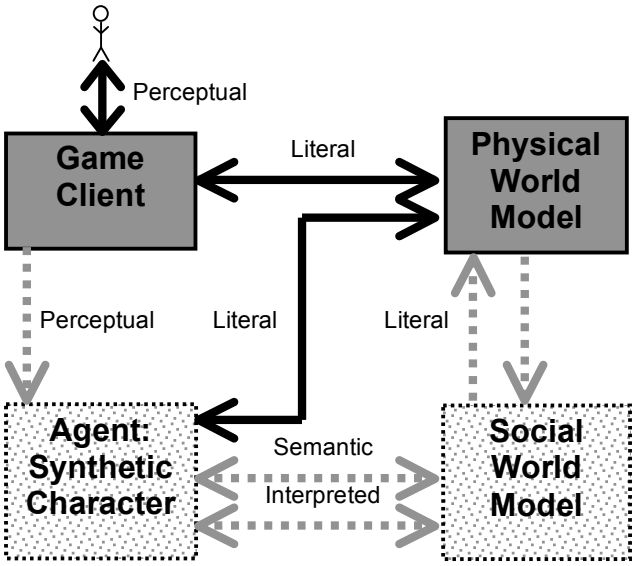

Figure 1: Levels passed among components

The primary task of the coupled components that sit between the synthetic characters and the game client is to transport the event descriptions from the characters or clients that create them to the ones that should know about them, dividing out levels according to what the receiving agent can handle. The downward arrow from the physical to the social model is to indicate that the locations of the characters' and human player's avatars matter in all but the most trivial virtual worlds. What can be seen and heard is location dependent (though it may only be represented topologically), and the flow of events has to reflect this.

The second task, reflected by the literal arrow pointing upward, is to provide translations for agents that only communicate at the semantic or interpreted level. The human behavior model driving a particular agent may be very rich, but it may operate at a level of granularity that is too coarse to provide animation instructions to its avatar (literal information). The social world model can be explicitly programmed to provide a mapping between that agent's interpreted output and animation that would reflect it. This is also the place to share a rich natural language capability that could take semantic-level information and render it as text or speech.

In summary, an agent's action is described by annotations at several different levels of abstraction simultaneously. There is no expectation that every agent will be able to understand or produce every level, and in some instances we arrange for our mediating components to fill in the missing information.

Using the Levels

As part of the DARPA-sponsored SCALE-UP (Social and Cultural Analysis and Learning Environment for Urban Pre- and post-conflict operations) project,[1] we recently demonstrated the feasibility of our framework and architecture. We developed a prototype system that integrated multiple agent controllers within a game environment so that they interacted with each other and with a human player. The system applied three of the abstraction levels discussed previously (the literal, semantic, and interpreted levels).

The SCALE-UP Project

Interest in and support for human behavior models has increased across all military services in recent years (Pew and Mavor, 1998). Recently, there has been increased interest in modeling the effect that emotions (Hudlika and Cañamero, 2004; Gratch and Marsella, 2004), beliefs, or cultural values have on human behavior. Additionally, models have been developed to describe crowd interactions (Silverman et al., 2002) and both combatant (Harper et al., 2002) and terrorist (Das et al., 2003) decision making. Some of these models have even been applied to increase realism and believability in military training experiences (van Lent et al., 2004; Toth, 2004). This increasing sophistication in modeling interpersonal interactions, including cultural or social factors, means that it is now possible and desirable to bring together state-of-the-art human behavior models developed through different approaches in a framework that facilitates interoperability, composability, and comparison. Demonstrating the viability of such a framework was the impetus for the SCALE-UP project.

SCALE-UP's goal is delivering light-weight, customizable, experiential training for missions that demand skillful navigation through social and cultural interaction. The military has a long history of using game-based simulations of the physical world (such as vehicle simulators, force-on-force strategic wargames, or first-person tactical games) to train for a variety of skills (Pappalardo, 2004). As missions increasingly require deft handling of interpersonal interactions, DARPA has recognized the need for game-based training that includes simulation of the social world. SCALE-UP focuses on the

[1] The work described here was sponsored in part by DARPA: contract #NBCHC050067.

need for reasonably convincing machine-controlled synthetic characters (here referred to as agents) so that users can practice and learn in a realistically populated simulated world without the need for large numbers of human actors.

We recently (December, 2005) completed a feasibility demonstration that used a game interface to allow a human user to play a *squad leader* who was trying to peacefully resolve a crowd control scenario populated with such agents. Because there are many aspects of human behavior, and many approaches to developing believable agents, and because different training scenarios will require agents that are specialized for certain roles or types of interpersonal interactions, an important part of the demonstration was to prototype a mechanism for integrating three different agents, each playing a different type of character in the scenario.

Figure 2: Squad leader addressing the crowd

One agent, Edutaniacs' PMFserv (Silverman 2001), controlled the *members of a crowd* that was expecting a food distribution. Another agent, CHI System's iGen/VECTOR (Zachary et al. 2001), controlled the *community leade*r who was the crowd's spokesman. The third agent, BBN's ENDER, which was developed as part of the project, handled an *agitator* who tried to influence the crowd against the squad leader (see Figure 2).

All of the agents were implemented in different software bases, and each had distinct needs for input and output data. Further, each model was designed to interact with the simulated world at different levels of abstraction. Based on our architecture, the interactions between the three agents and the human player all used a three-level message format consisting of the literal, semantic, and interpreted levels.

Examples of the Levels in SCALE-UP

The agent used to control the crowd members in the scenario specialized in modeling the likelihood for an individual to support or oppose government authority, depending on his reaction to events, personal needs, values, and tendencies. This agent modeled a person's internal mental and emotional state with a lot of detail, but handled understanding external events and specifying the person's resulting actions more abstractly at just the interpreted level. Because the simulation framework (the coupled world models) was programmed to understand the various actions in the scenario and "fill-in" the needed information at the other levels, the crowd members were able to react to overhearing the dialog between the squad leader (human player) and the community leader.

The crowd perceived this conversation as a series of events, each described in the interpreted level as indicating different levels of respect, security, or food. For example, if the squad leader chose to bow towards the community leader, the simulation system filled in the event message's interpreted level to contain the description "respectful" in the respect element, "neutral" in the security element, and "unknown" in the food element.

The literal level was used primarily for directing the game client's visual rendering. The simulation framework was able to first map a crowd member's interpreted level actions onto literal level actions, and then use those actions to request specific perceptual level rendering. For example, if one of the crowd members expressed a particular level of grievance, the event message would be filled in with a literal level action such as PoundFist, and the framework would then request the corresponding game animation.

The literal level was also used by the agent that controlled the community leader character. This agent could recognize specific actions of the squad leader, such as selecting a specific dialog choice from a menu, by examining the text portion of the literal level of the event message. This character could also generate specific literal level behaviors, such as IdleCrossArms, in reaction to a change in the agent's internal emotional state becoming angry.

The semantic level was only used by the agent that controlled the agitator character when it needed to handle a part of the example situation that required deeper domain knowledge. The scenario allowed the squad leader to choose to distribute MREs (meals-ready-to-eat) to the hungry crowd. Because the semantic level description of this action contained information about the type of food in the MRE, the agent controlling the agitator was able to recognize that it contained an ingredient that was religiously prohibited. Unlike the other two other agents, this one had only a minimal mental and emotional state, but did have the domain knowledge needed to reason about halal food, and to interpret the distribution of non-halal food as disrespectful. The agitator's reaction upon recognizing this disrespectful act was to tell the crowd members about it at the interpreted level.

Lessons Learned

Our experience with the SCALE-UP feasibility demonstration showed that different, independently developed systems for generating computer controlled behavior could play in the same scenario when their interaction is facilitated by a simulation system using messages with multiple abstraction levels to communicate between characters. The inter-character communications consisted of the three intermediate levels: literal, semantic, and interpreted. The perceptual level was only generated by the game's visual rendering for the human participant, while the narrative level was not used in this small scenario. The simulation framework's ability to translate between abstraction levels enabled the characters controlled by different agents to react to each other and to the human controlled squad leader. The agents focused on modeling different aspects of human behavior, yet were able to interoperate through multiple abstraction level messages.

Conclusion

As games become more complex, game developers for better or worse will be faced with the same problems that we have encountered in our work. Getting realistic social behavior in convincing detail, especially if it involves natural language, will inevitably lead to incorporating heterogeneous sets of agents into games. These agents will likely be developed by people with different scientific and engineering backgrounds, and will have different strengths and weaknesses. Limitations in time and resources will mean that the game framework will have to bend to fit the interface limitations and requirements of these agents, rather than the other way round.

Our experience creating the SCALE-UP feasibility demonstration has made it clear that independently developed agents, because of their different concerns and internal designs, cannot easily be made to understand the same description of an event – one size does not fit all. The three-level event descriptions that we developed and used in our demonstration were crucial to our own problems of incompatible "impedance matches" among our three agents. We believe that adding the perceptual level at the bottom clarifies some conceptual questions about what the literal level should be taken to represent, and that the narrative level makes plain the need for explicitly represented overall thematic control in story telling. In our ongoing and future work, we will be concentrating on the social aspects of this work. We hope to develop a social engine that can "fill in the blanks" for simple agents that need to work in the same game contexts as agents that are already socially sophisticated.

References

Das, S., Ruda, H., and Zacharias, G. "Predicting Terrorist Actions Using Sequence Learning and Past Events," Proceedings of SPIE, Volume 5071, AeroSense, Orlando, FL (April), 2003.

Gratch, J., Marsella, S., "Evaluating the modeling and use of emotion in virtual humans," Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, New York, New York, 2004.

Harper, K., Aykroyd, P., Zacharias, G., Middleton, V. Hannon, C., "Cognitive Modeling of Individual Combatant and Small Unit Decision-Making within the Integrated Unit Simulation System," Proceedings of the 11th Conference on Computer-Generated Forces and Behavior Representation, 2002

Hudlika and Cañamero, "Architectures for Modeling Emotion: Cross-Disciplinary Foundations", 2004 AAAI Spring Symposium

Pappalardo, Joe. "Demand for Non-Combat Skills Fuels Interest in Games" National Defense, 2004.

Pew, R. W. and Mavor, A., (Eds), Modeling Human and Organizational Behavior: Applications to Military Simulations, Washington, D.C., National Academy Press, 1998.

Silverman, B., More Realistic Human Behavior Models for Agents in Virtual Worlds: Emotion, Stress, and Value Ontologies (Technical Report), 2001.

Silverman, B.G., Johns, M., O'Brien, K., Weaver, R., & Cornwell, J. (2002b). Constructing virtual asymmetric opponents from data and models in the literature: Case of crowd rioting. Proceedings of the 11th Conference on Computer Generated Forces and Behavioral Representation, Orlando, Florida, 97-106.

Silverman, B.G., Johns, M., O'Brien, K., Weaver, R., Cornwell, J., "Constructing virtual asymmetric opponents from data and models in the literature: Case of crowd rioting", Proceedings of the 11th Conference on Computer Generated Forces and Behavioral Representation, 2002, Orlando, Florida, 97-106.

Toth, J.A., "After-Action Report: Integrating disparate human behavior representations in a first-person shooter game based on the Blackhawk Down incident in Mogadishu." (IDA Document D-2975), Alexandria, VA: Institute for Defense Analyses, 2004.

van Lent, M., McAlinden, R., Brobst, P., et al., "Enhancing the behaviorial fidelity of synthetic entities with human behavior models", 13th Conference on Behavior Representation in Modeling and Simulation (BRIMS), SISO, May 12-15, 2004.

Zachary, W., Santarelli, T., Ryder, J., Stokes, J., & Scolaro, D. (2001). Developing a multi-tasking cognitive agent using the COGNET/iGEN integrative architecture. In Proceedings of 10th Conference on Computer Generated Forces and Behavioral Representation, (pp. 79-90). Norfolk, VA: Simulation Interoperability Standards Organization (SISO).