Interoperable Human Behavior Models for Simulations

David McDonald
Richard Lazarus
Alice Leung
Talib Hussain
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
617-873-8000
{ dmcdonald, rlazarus, aleung, thussain }@bbn.com

Gnana Bharathy
Roy J. Eidelson
Nuria Pelechano
Evan Sandhaus
Barry G. Silverman
University of Pennsylvania
Philadelphia, PA 19104-6315
215-573-8368
{ bharathy, npelecha, esandhau, barryg } @seas.upenn.edu, royeidel@psych.upenn.edu

Keywords: Human Behavior Models, Interoperable Agents, Simulation-Based Training, Game-Based Training

ABSTRACT: *Modern simulations and games have limited capabilities for simulated characters to interact with each other and with humans in rich, meaningful ways. Although significant achievements have been made in developing human behavior models (HBMs) that are able to control a single simulated entity (or a single group of simulated entities), a limiting factor is the inability of HBMs developed by different groups to interact with each other. We present an architecture and multi-level message framework for enabling HBMs to communicate with each other about their actions and their intents, and describe the results of our crowd control demonstration system which applied it to allow three distinct HBMs to interoperate within a single training-oriented simulation. Our hope is that this will encourage the development of standards for interoperability among HBMs which will lead to the development of richer training and analysis simulations.*

1. Introduction

The rich human behavior models (HBM) that have been developed in the psychological, physiological, or political science communities, if they are used at all in a modern game or simulation, will control only one synthetic entity (or coordinated group of entities) and will only produce behaviors whose expression and intent are targeted at the human interacting with the system.

However, in the applications of the future, we believe that there will be a need to draw on HBMs with complementary skills or specialties that have been developed by different groups, and consequently that the HBMs will be required to leave their solipsistic self-contained worlds and communicate their actions and intent to other HBMs controlling other synthetic characters.

Given the differences in scope and focus of the HBMs that we have looked at, and to highlight what the extensive use of HBMs in simulations could achieve, we will refer to them in this paper as 'people engines' (PE).

In a recent DARPA-sponsored project,[1] we developed a prototype of SCALE-UP (Social and Cultural Analysis and Learning Environment for Urban Pre- and Post-Conflict Operations), a computer-based training and analysis system. The goal of SCALE-UP is to provide a training and analysis capability to enhance force effectiveness in urban settings through the use of multiple HBMs to control avatars within a simulated environment.

In the project, we developed a framework for enabling distinct people engines to communicate about their actions and intents. Using this framework, we developed a prototype test bed that integrated three heterogeneous people engines with a massive multiplayer on-line game for visualization. Two of the people engines were based upon existing HBMs (Silverman et al., 2002, Zachary et

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

al. 2001), and a third was developed as part of the project. To explore the benefits of our approach, we developed a specific training scenario in which a human player was given the goal of dispersing a crowd through a sequence of dialog choices. The demonstration was based on lessons learned for dealing with a crowd with Iraqi cultural tendencies. The avatars in our system (a village leader, a crowd of twenty individuals, and an agitator) were controlled by different people engines that reacted to the actions of the human player and of other avatars based on their social and cultural interpretations of those actions.

In this paper, we present the messaging framework that we developed, describe the architecture of the SCALE-UP test bed, introduce the details of our demonstration scenario and present the outcomes. We conclude with our thoughts on the next steps required to achieve effective standards for interoperability among PEs that will both drive the development of future PEs, as well as accommodate the rich variety of human behavior models that currently exist.

2. Motivation: Interoperable People Engines

In a typical game or simulator, people engines (PEs) need to pass along information about one or more of three kinds of behaviors: things that they say, gestures that they make, and/or actions that they take. Sophisticated PEs need to understand the events in the game sufficiently that they can determine their response or general activity without the use of predefined scripts. The interoperability issues arise because different PEs will often operate at different levels of abstraction, both when receiving information about the world from other PEs and for conveying information about their own activities.

Each of the PEs that we worked with had its own model of perception and acting. Some responded to very abstract events like being insulted; some to more concrete events like being physically approached. Some produced concrete action directives such as uttering the words "Go away;" others produced 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 translate incorrectly between levels. For instance, one culture's sign for success (e.g., thumbs-up) can be another's obscene insult. A person that is unaware of this cultural difference in how the literal action is translated may inadvertently give offense or misinterpret someone else's action. We want to be able to incorporate the possibility for this kind of cultural mistake in our training systems, which means that the language that is used to describe a PE's actions must not preclude the possibility of different PEs giving different interpretations to the same event.

In a game or a simulator where avatars only interact with human players, there is usually no significant need to convey information about the intentions behind the behaviors of the PEs – the human players are expected to make their own interpretations. However, in order to enable different PEs to interact with each other as well as with humans in a virtual environment, it is important to provide those PEs with the capability of understanding and behaving at different levels of abstraction. In both cases, though, the virtual world simulator needs concrete descriptions of what the PE is doing that it can understand and that rendering clients can turn into animations and sound.

As part of our SCALE-UP effort, we developed a five-level language for describing agent behaviors. Each level is a valid description of what was done. 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 a PE takes is encoded in terms of this language, and every PE sees the actions of the other PEs via the messages sent from them and encoded in the multi-level vocabulary.

We believe that, in the long-term, such a language may form the basis for a messaging standard that enables enhanced interoperability among human behavior models for controlling synthetic agents in games and simulators. We hope that such a standard will drive the development of a new class of human behavior model that is designed to be cooperative rather than stand-alone.

3. Heterogeneous Messaging through Multiple Levels of Abstraction

In this section, we describe four of the five levels in our framework, from the most concrete to the most abstract. (The fifth level, narrative, was not directly addressed by the SCALE-UP effort, see McDonald et al., 2006) In the following sections, we show how these levels are mapped to communications between the PEs, 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 game or simulator is "represented" by the audio and video that a rendering client can produce. Utterances are sound, gestures are sets of pixels 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, but 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.

Level 2: Literal

At the *literal level*, the flow of time and activity in the simulator is divided into discrete 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 four.

At the literal level, utterances are represented as a single event annotated with the speaker and the sequence of words and (if possible) prosodic information, 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 PEs). The social model may also annotate events at the interpreted level by providing cultural or context specific interpretations of events (or possible responses, etc.) as described briefly in section five.

Figure 1 illustrates an example message that would be communicated to a PE to describe a specific action that has happened. The message contains information about the specific act (i.e., making the demand "You must leave the area!"), a representation of its meaning, as well as an interpretation of the effect of the action on the crowd it is addressed to (i.e., to show disrespect).

```
<action>
     <type>external to crowd interaction
space</type>
     <by>squad-leader</by>
     <to>crowd-all</to>
     <literal-level>
          <text>You must leave this area!</text>
     </literal-level>
     <semantic-level>
          <act>demand</act>
          <proposition predicate=”required”>
              <argument>
                 <proposition predicate=”leave”>
                    <arg agent=”addressee” \>
                    <arg location=”square” \>
                 </proposition>
             </argument>
          </proposition>
          <affect>command</affect>
     </semantic-level>
     <interpreted-level>
          <effect>
               <on>crowd-all</on>
               <content>
                    <respect>disrespectful</respect>
               </content>
          </effect>
     </interpreted-level>
</action>
```

Figure 1: Sample communication about an action

4. Coupled-Worlds Architecture

To support communication among PEs and between the PEs and the interface to the human players, we developed the coupled-worlds architecture shown in Figure 2. 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 dark solid lines for components and links that one would expect in a "conventional" game, and light gray dotted lines for what we have added.

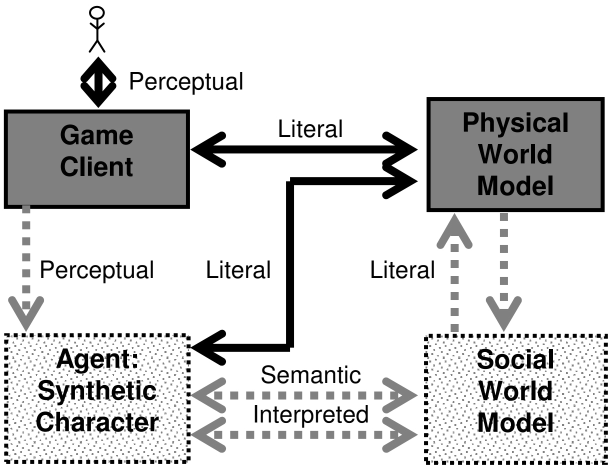

Figure 2: Levels passed among components

On the top left, we have the game client. This is where human users access the scenario. The 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 dotted line linking it to the synthetic characters is to remind us of the possibility that some PEs might want the raw data and that 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 PEs. 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.

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 PEs that only communicate at the semantic or interpreted level. The PE driving a particular avatar may be very rich but 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 a 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.

5. Crowd Control Scenario

To understand the interoperability issues and then demonstrate the feasibility of the messaging framework that we designed to address it, we applied our architecture to create a training scenario that integrated three distinct human behavior models as people engines, and used the Big World[2] game environment system to render the virtual world and portray the behaviors chosen by the people engines.

In our training scenario, a human user plays a *squad leader* who is trying to peacefully resolve a problem with a crowd that is populated with synthetic characters. Three different people engines were used, each playing a different character (or character type) in the scenario. One people engine, Edutaniacs' PMFserv (Silverman 2001), controlled the *members of a crowd* that were expecting a food distribution that had not arrived. Another PE, CHI System's iGen/VECTOR (Zachary et al. 2001), controlled the *community leade*r who was the crowd's spokesman. The third PE, BBN's ENDER, which was developed as part of the project, modeled an *agitator* who tried to influence the crowd against the squad leader. Figure 3 shows the crowd scene.

The people engines were implemented using different software bases, and each had distinct needs and capacities for input and output. For the most part, each PE was designed to interact with a simulated world at different levels of abstraction. The interactions between the three PEs and the human player all used a three-level message format consisting of just the literal, semantic, and interpreted levels. The perceptual level was provided by Big World for the benefit of the human player.

The possible actions of the person playing the role of the squad leader were strictly limited to navigating a fixed dialog tree of utterances and accompanying actions. (This dialog tree also served to bound the range of actions that the developers of the PEs had to consider when adapting their systems to our scenario. However, with one exception, their actions were not scripted.) We created a fully filled-in message for each of the options that the

[2] "Big World" is a trademark of BigWorld Pty Ltd

person could choose, and issued it to all of the PEs at the moment the choice was made. (Everyone in the scene is assumed to be able to hear the entire conversation and to see all gestures made by any other character.)

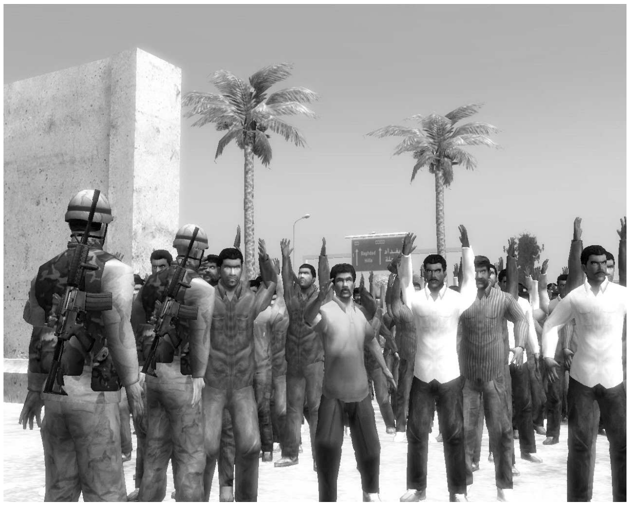

Figure 3: Squad leader addressing the crowd

The natural language responses made by CHI System's community leader were also scripted by the dialog tree. CHI's PE used the literal level of the messages from the squad leader in order to recognize which action was taken. The unscripted emotional and cognitive reactions of this PE led to specific literal-level behaviors in the messages that it sent out: for example IdleCrossArms, to reflect the change in the PE's internal emotional state to becoming angry.

CHI's PE could move its avatar so that it could face the crowd when addressing it at the beginning of the scenario and then turn to face the squad leader when he arrived on the scene. The PE could direct its avatar to approach the (avatar of the) squad leader and had a tuned animation that let it walk up to the squad leader and stand at the normal face-to-face speaking distance in its culture, which is closer than most American's find comfortable. This provided one of the training opportunities within the scenario. How the person making the choices for squad leader reacts in this instance, as well as others such as the choice to distribute pork-filled MREs (meals-ready-to-eat) discussed just below, is an example of the kind of actions that distinguish a naïve player from an experienced one.

The semantic level was used by BBN's PE (the agitator character) when it needed to handle an action by the squad leader that required deeper domain knowledge in order to understand its cultural implications. The scenario allowed the squad leader to choose to distribute MREs to the hungry crowd. Because the semantic-level description of this action contained additional information about the type of food in the MRE, the BBN's PE was able to examine this information and recognize that the MRE contained an ingredient that was religiously prohibited. Unlike the other two other PEs, this PE 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. At that point, it also used the literal level to change its 'idle state' (the animation sequence that its avatar takes between explicit commands) to reflect it increased anger and to simulate the effect of talking to its neighbor avatars by moving its avatars head from side to side.

The twenty avatars that made up the crowd were individually controlled by a single instance of the PMFserv people engine. Silverman and his group at the University of Pennsylvania have had extensive experience in modeling crowds for computer simulation by PMFserv (see, e.g., Silverman et al. 2006a,b). In the crowd model used in this project, we drew on the work by Eidelson (2003) on 'Dangerous Ideas'. In brief, this model says that a person's affinity with a group and the possibilities of conflict within it or the propensity of individual members to stay or leave depends on how the member see themselves and the others in terms of five modeled beliefs: vulnerability, injustice, distrust, superiority, and helplessness.

The model of these beliefs in PMFserv depended on the values that individual crowd members (usually clones of several archetype members) had in their goals, standards, and preference trees. (See Silverman & Bharathy 2005 for and example of how these 'GSP' trees are used. See Ortony, Clore, and Collins 1988 for the original development of many of the ideas embodied in the GSP trees of PMFserv.) The crowd members' perception of the events in the scenario and their reactions to it are all filtered through their present emotional and cognitive state and combined with the other performance modulator functions deployed for this crowd model to arrive at the expected utility of the various actions they could take.

Since the crowd has been modeled at a very abstract level when compared with the PEs controlling the other two synthetic characters, PMFserv viewed the conversation between the squad leader and the community leader in appropriately abstract terms represented at the interpreted level. All of the literal and semantic content of the events in the conversation were projected to the interpreted level in terms of Maslow's hierarchy of needs (life for a villager can be a day-to-day struggle) and rendered as 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.

Similarly, the output actions of individual crowd members were summarized by PMFserv as their ‘grievance state’, given on a scale from -4 to +4. This, in turn, was translated by the simulation framework (the coupled-world models) which filled in the literal level of messages from the individual members of the crowd. This use of the literal level was primarily for directing the game client’s visual rendering so that the human player could understand the state of the crowd in natural way (their reaction to the action the person had just made and the collective effect of his actions so far). The simulation framework first mapped a crowd member’s interpreted level grievance value onto literal level actions, and then used those actions to request specific perceptual level rendering. For example, a particular level of grievance in a crowd member’s event message might lead the social world model to fill in the message’s literal-level with an action such as PoundFist, and the physical world model would then request the corresponding game animation.

6. Demonstration

We envision that future versions of SCALE-UP will be used for experiential, game-based training systems that allow human learners to interact with a variety of computer controlled entities driven by different people engines. In order for students to learn from scenarios populated by people engines, the characters encountered must react appropriately to the student’s actions or decisions. In particular, culturally insensitive actions should have a visible, negative impact on the student’s ability to succeed in their mission. Additionally, since the student may need to learn about the differences among several populations, any scenario should support being populated with sets of characters who will react differently to the same series of student actions.

Thus, for the SCALE-UP demonstration, we wanted to show that a series of “naïve” decisions would have a less successful outcome than a series of more culturally informed actions, where the outcome is essentially the attitude and reactions of the synthetic characters. We also wanted to show that either series of decisions would have a different impact on crowds consisting of more hostile or more moderate individuals. Table 1 shows the different quantitative outcomes that we wanted to demonstrate.

In our SCALE-UP demonstration, we selected two distinct paths (sequences of dialog choices) through the scenario. One path was chosen to reflect a player who understands how to deal with an Iraqi crowd and community leader. The other path represented a player who was inexperienced in dealing with such a scenario and naïve about the crowd’s culture. We then instantiated two versions of the playable scenario, each populated with a different set of crowd members. One crowd was tuned to have moderate views towards Americans and the other had more extreme (negative) views towards Americans and a higher propensity towards violence.

Our focus was on producing differentiated outcomes and not on accurately modeling a specific culture or situation. Though limited to a relatively small set of gestures, the crowd avatars were able to visually show a nearly violent reaction by starting to make throwing motions. Hostility was expressed through vigorous arm waving. Similarly, distrust was expressed through crossing arms, and agreement through nodding. Because the individuals in the crowd were parameterized slightly differently, they did not execute these gestures all at the same time, thus producing a more realistic scene. In addition to the visibly differentiated crowd behavior that we observed through the game environment, we examined the internal state of the crowd members to verify the differentiated outcomes of each set of conditions. The average grievance state of the crowd members could be tracked throughout the short scenario. This measure of the crowd’s mood could be used as a measure of the student’s success in convincing the crowd to disperse.

Table 1: Crowd reactions to series of decisions

	Moderate crowd	**Extreme crowd**
Culturally naïve actions	Hostile, but not violent	Violent or nearly violent
Culturally aware actions	Cautious, cooperative	Hostile, but not violent

7. Lessons Learned

We note several issues that arose in the course of implementing a messaging system with multiple levels of abstraction between different people engines.

First, training systems such as the SCALE-UP prototype, which are made up of multiple people engines and a game client as the human interface, benefit from using a physical world model outside of the game. Originally, we intended to model some aspects of the physical world which were not covered by the game itself, such as “off-stage” character movements, separately from the game’s physical world simulation. However, it became clear that there were also benefits to using a separate, more abstract, physical world model to represent things that the game’s world model already covered. For example, BigWorld, like most game engines, represents physical locations using a coordinate system. Specific physical coordinates in the scenario map were associated with a list of more abstract notional locations, such as “front-of-crowd” or “home”. The people engines selected movement actions based on the notional locations. Thus, when the absolute locations of the avatars was changed, the people engines still behaved correctly; only the mapping between absolute coordinates and notional locations needed to be updated. This notion of separating the social and physical world model from the game client should also allow the people engine testbed to use different game clients as appropriate. For example, the BigWorld game client was suitable for the first-person scenario demonstrated, but a more strategic level game client might be better for other types of scenarios. Since most game clients will not be designed to interface with external people engines, it is likely that each new game client will require a custom interface to the rest of the testbed.

Second, the issue of the timing of people engine reactions was significant. If a character reacted too quickly or too slowly, it detracted from the player's immersion into the scenario. Game clients do not always provide notification of when an animation or movement has been completed, and computationally intensive people engines will take varying amounts of time to generate a reaction. In our infrastructure and message system, we used explicit event duration and notification delays to tune the scenario and prevent "faster than light" reactions. Yet using explicit notification delays (so that people engines won't "notice" an action until a person would have noticed it) can make it harder for a computationally intensive people engine to produce a timely reaction. Distributing the computational load among more computers can minimize the problem, but it is also likely that very computationally intensive people engines will not be suitable as part of a real-time training scenario.

Finally, there were questions about how to use human behavior models that were designed to be brains rather than bodies in a system that allowed a human participant to constantly view the other characters. Many cognitive or emotional models operate at a high level of abstraction and are not concerned with lower level actions such as smiling or shifting their weight from one foot to the other. We used a translation layer that turned emotional state into visible behaviors expressing that emotion. However, there were still questions about whether these generated behaviors should be one time events (cross arms, pause, then return arms to sides) or new default postures (cross arms, and leave them crossed until doing something different). Various people engines can also differ in whether they report emotional state in response to notifications of events, at periodic intervals, or upon a change in state. We chose to translate the emotional state of the crowd members, reported in response to dialog events or other actions, as new default postures. However, the people engine controlling the community leader instead chose to initiate one time events upon a change in emotional state. Both methods led to reasonable looking characters.

8. Conclusions

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, facilitated by a simulation system using messages with multiple abstraction levels to communicate between characters (human and synthetic)

As game- and simulation-based training systems become more complex, system developers, for better or worse, will be faced with the same problems that we encountered in our work. Achieving a realistic level of social behavior with convincing details, especially if it involves natural language, will inevitably lead to incorporating heterogeneous sets of PEs into such systems. These PEs will likely be developed by people with different scientific and engineering backgrounds, and 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 PEs, rather than the other way round.

To further enhance and validate our ideas, we plan to develop an enhanced test bed for model comparison and validation using a more complex virtual world. We plan to define a set of challenge problems and draft interchange standards to focus groundbreaking behavior modeling research in the domain of socially and culturally affected interpersonal interactions. Through these steps and more, we hope to work towards establishing a robust communication framework and architecture that will lead to improved cooperation among HBMs and enhanced training and analysis applications.

9. Acknowledgements

The authors would like to acknowledge the contributions of the following individuals towards the SCALE-UP vision and demonstration: Ralph Chatham (DARPA), Thomas Santarelli and Larry Rosenzweig (CHI Systems), David Nielsen and Keith Copenhagen (Total Immersion Software), Rich Shapiro, Clint Hyde, William Ferguson, and Rick Allan (BBN Technologies).

10. References

Eidelson, R.J. & J.L. Eidelson "Dangerous Ideas: Five Beliefs That Propel Groups Toward Conflict", American Psychologist, 2003.

Ortony, A., G. Clore, and A. Collins "The Cognitive Structure of Emotions", Cambridge University Press, 1988.

McDonald, D., A. Leung, W. Ferguson, and T. Hussain "An Abstraction Framework for Cooperation

Among Agents and People in a Virtual World", Proceedings of Artificial Intelligence and Interactive Entertainment Conference AIIDE-06, June 2006.

Silverman, BG, Johns, M, Cornwell, J, O'Brien, K, "Human Behavior Models for Agents in Simulators and Games: Part I – Enabling Science with PMFserv," PRESENCE, April 2006a.

Silverman, BG, Bharathy, G, O'Brien, K, Cornwell, J, "Human Behavior Models for Agents in Simulators and Games: Part II – Gamebot Engineering with PMFserv," PRESENCE, April, 2006b.

Silverman, B., and G. Bharathy "Modeling Cognition and Personality in Leaders", BRIMS05, May 2005.

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).

Author Biographies

DAVID McDONALD a Senior Scientist at BBN Technologies. He is a computational linguist with interests in conceptual modeling and knowledge representation. He is applying his skills in projects to understand notices to airmen (NOTAMS) and as part of AQUAINT.

RICHARD LAZARUS is a Director of Research at BBN Technologies. He manages a group pursuing research in scalable architectures, distributed and experiential learning, and knowledge representation and visualization. Also, he is the program manager for BBN's DARWARS program.

ALICE LEUNG is a scientist at BBN Technologies. Her current interest is in harnessing games for behavior research and training. Dr. Leung was the technical lead for the SCALE-UP project. She is also a co-Principal Investigator on the DMSO/AFRL/ARL SABRE (Situation Authorable Behavior Research Environment) project. SABRE is designed to allow researchers to study the effects of culture on team communication and decision making within a controlled, game-based scenario.

TALIB HUSSAIN is a Senior Scientist at BBN Technologies. He works on game-based training, evolutionary algorithms for robust tactical behaviors, and machine learning. He is currently technical lead on the DARPA-sponsored POIROT project, which is developing the capability for integrated learning of tasks based on a single observation of a human performing the task. He is also a technical contributor on the DARPA-sponsored ADROIT project, in which he is developing cognitive control capabilities for networks based on software-defined radios.

GNANA BHARATHY is a Doctoral Candidate in Systems Engineering with Professor Silverman at the University of Pennsylvania, where he is putting final touches to his dissertation in agent based human behavior modeling. His dissertation research on the same topic is a recipient of awards from Int'l. Council on Sys Eng (INCOSE) and Wharton Risk Management Center. Gnana also serves at Ackoff Center in the capacity of research associate. Previously, Gnana had several years of experience in risk, environmental and technology consulting.

ROY J. EIDELSON is a clinical psychologist and executive director of the University of Pennsylvania's Solomon Asch Center for Study of Ethnopolitical Conflict. His research interests focus on the similarities and differences between the conflict-engendering core beliefs of individuals and the collective worldviews of groups.

NURIA PELECHANO is a Fulbright Doctoral Candidate in Computer and Information Science with Professor Norman Badler and a research assistant for the Building Simulation Group at the University of Pennsylvania. Her research focuses on crowd simulation dealing with both the low level details of realistic human movement and the high level wayfinding and decision-making based on roles and communication.

EVAN SANDHAUS is the Lead Software Developer at the Ackoff Center for Advancement of the Systems Approach. Mr. Sandhaus holds degrees in Computer Science from both Williams College and Villanova University and is currently studying at the University of Pennsylvania. His research interests include intelligent agents, behavior modeling, decision simulation, word sense disambiguation, and machine learning.

BARRY SILVERMAN is a Professor at the University of Pennsylvania, where he is the Director of the Ackoff Center for Advancement of the Systems Approach. His research focuses on aesthetic and cognitive engineering of embedded game-theoretic agents that can help humans improve their learning, performance, and systems thinking in task-environments.