SANDIA REPORT

SAND2009-0840
Unlimited Release
Printed February 2009

Enabling Immersive Simulation

Robert G. Abbot, Justin D. Basilico, Matthew R. Glickman, Derek Hart, Michael Mateas, Josh McCoy, and Jonathan H. Whetzel

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2009-0840
Unlimited Release
Printed February 2009

Enabling Immersive Simulation

Robert G. Abbot, Justin D. Basilico, Matthew R. Glickman, Derek Hart,
and Jonathan H. Whetzel
Cognitive and Optical Military Systems, Cognitive Systems Research & Applications
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1004

Michael Mateas and Josh McCoy
Expressive Intelligence Studio
University of California Santa Cruz
1156 High Street
Santa Cruz, CA 95064

Abstract

The object of the "Enabling Immersive Simulation for Complex Systems Analysis and Training" LDRD has been to research, design, and engineer a capability to develop simulations which (1) provide a rich, immersive interface for participation by real humans (exploiting existing high-performance game-engine technology wherever possible), and (2) can leverage Sandia's substantial investment in high-fidelity physical and cognitive models implemented in the Umbra simulation framework. We report here on these efforts. First, we describe the integration of Sandia's Umbra modular simulation framework with the open-source Delta3D game engine. Next, we report on Umbra's integration with Sandia's Cognitive Foundry, specifically to provide for learning behaviors for "virtual teammates" directly from observed human behavior. Finally, we describe the integration of Delta3D with the ABL behavior engine, and report on research into establishing the theoretical framework that will be required to make use of tools like ABL to scale up to increasingly rich and realistic virtual characters.

CONTENTS

FIGURES

NOMENCLATURE

ABL	A Behavior Language
ARTS	Augmented Reality Training System
DOE	Department of Energy
NPS	Naval Postgraduate School
SNL	Sandia National Laboratories

1. INTEGRATION OF THE UMBRA SIMULATION FRAMEWORK WITH THE DELTA3D GAME ENGINE

Umbra

Originally developed for simulating robotics applications, Sandia's Umbra (Gottlieb, McDonald, Oppel, Rigdon, & Xavier, 2002) modeling and simulation framework has served as a fertile platform for the development of a variety high-fidelity physical as well as cognitive models. Via Umbra's flexible, modular architecture, such models may be composed to yield higher-level system models, and the interactions among heterogeneous collections of such models may be simulated simultaneously spanning a broad range of phenomenological domains (e.g. radiation, etc.). Users can view representations of these models in 3D virtual environments as well as manipulate and interact with the models in real-time.

Umbra does provide several of the features typically associated with modern game engines, such as a graphics engine, a scripting language, character animation, collision detection, and input device integration. Nonetheless, with few exceptions the role of a human user with respect to an Umbra application is typically that of an analyst, configuring and controlling a simulation and observing the results. To be able to fully leverage the Umbra's high-fidelity models in an application that provides a rich, immersive interface (such as a virtual training system), additional capabilities are required. One such requirement is the ability to generate spatially-located sounds. Another significant requirement is a graphical level editor, which permits drag-and-drop composition of rich visual scenes.

Umbra & Delta3D

The additional capabilities necessary for using Umbra models in conjunction with an immersive, participatory interface are provided via integration with the open-source Delta3D (McDowell, Darken, Johnson, & Sullivan, 2005) game engine. Delta3D was developed by the MOVES Institute at the Naval Postgraduate School (NPS).

From the Umbra-developer perspective, Delta3D appears an optional front-end for Umbra simulations providing complete game-engine functionality. From the perspective of Delta3D developers, Umbra can appear as an optional, sophisticated simulation engine/library available via Delta3D. Umbra also offers Delta3D developers the opportunity to leverage many high-fidelity physics-based models developed within the Umbra framework. Specific game-engine functions now available to Umbra developers include: improved visual effects (e.g. richer lighting options, weather effects, editable particle effects), improved performance (e.g. BSP culling), 3D sound, game networking, a client/server architecture, and a level editor.

1.2.1. Technical Approach

A multi-threaded approach was used to integrate the two environments. The multi-threaded approach allows Umbra's high fidelity simulation models to run without having an adverse affect on the frame rate being displayed in the Delta3D visualization window. Since the goal of this project was to develop a game engine interface, a constraint of real-time responsiveness was deemed to be very important. If the high fidelity Umbra models run within the same thread as the visualization, a noticeable degradation in visualization rendering will occur.

With the multi-threaded approach came the issue of synchronizing the data being modeled in Umbra with the data being visualized in Delta3D. To confront this issue, a lockable data structure was chosen to keep the data on each side of the thread boundary synchronized. The Delta3D side will try to get an updated version of the state data each time it renders a frame. If the lock is currently held by the Umbra side, it will simply render the data it internal and try to get the lock during the next frame. The Umbra side will always be updating the data, and Umbra will wait for the lock as long as it takes for the lock to become free. Umbra is the data-provider in the simulation. The Delta3D side is mostly setup to visualize the data. However, the design of the shared data structure allows for bi-directional data flow. For instance, the DI-Guy Character Animation Library is running as a component of Delta3D; however, for some of our sensor models running in Umbra, we need positions of the avatars being modeled using DI-Guy. In this case, Delta3D posts data to the data structure, and Umbra uses data to update its sensor models.

Using the shared data structure, Umbra's capabilities have expanded to include components previously not available. Delta3D has sound and 3D user interface components that are now accessible to Umbra users using this integration effort. In order to make the design applicable to both existing and new projects, the program runs much the same way as a typical Umbra application. Within Umbra you can create sound, user interface, geometry, and DI-Guy proxy modules that hold the synchronized data. For sound, the user interface, and geometry, we leveraged components that already existed within Delta3D. This allowed us to expand Umbra's capabilities without having to integrate and implement a variety of new software packages.
To integrate Umbra with Delta3D, a design was chosen that takes advantage of Umbra's World module design paradigm and Delta3D's Game Manager. Umbra's World modules allow you to add a group of similar modules under the umbrella of a managing "world" module. So on the Umbra side, a world is created for each type of data that needs to be synchronized (sound, user interface, geometry, and DI-Guy). The world then allows creation of specific instances of each type of data, like a sound or a sphere geometry. The worlds will update at the end of the Umbra update loop, after all their children modules have updated their data, such as position and other state information. The worlds will iterator through each child module and update the shared data structure if the data in the child module has been updated. The world design makes it simple for the user to manage synchronization of data across the thread. On the Delta3D side, a GameComponent does much the same job. It collects all the GameActors for each type of component (again sound, user interface, geometry, DI-Guy), and updates them, if necessary, each time a frame is rendered. The other benefit using the Game Manager is that it allows us to leverage the scenario design tool, STAGE, that Delta3D provides. The Game Manager can load scenarios created in STAGE and create the necessary GameComponents and GameActors, which in turn create their Umbra-side proxy data objects. When the proxy data object is created on the Umbra side, the shared data structure is notified to tell the Delta3D side the a concrete data object needs to be created. The reverse is true also: if Delta3D adds an Actor, the shared data structure is informs Umbra that a new child module needs to be created.

2. TRAINABLE AUTOMATED FORCES IN UMBRA

2.1. Introduction

A strategic goal for this project aimed at forming a bridge between Sandia' modeling and simulation community and research conducted under Sandia's Cognitive Science & Technology business area to create more realistic, user-focused interactive simulations through the analysis of human behavior in these environments. Through this combination, the project would provide Umbra with unique capabilities such as generating non-player character (NPC) behaviors through user demonstration, as opposed to traditional costly methods of knowledge elicitation and programming, and providing new methods for measuring user performance and debrief through behavior model comparison.

Trainable Automated Forces (TAF) (Abbot and Basilico 2008) is the technical project focusing on the behavior generation through user demonstration and Automated Expert Modeling and Student Evaluation (AEMASE) (Abbot 2006) is the project focusing on user performance analysis through behavior model comparisons. Both of these technologies are built upon a common software framework created for the creation of human behavior models, colloquially referred to as the Cognitive Foundry (Basilico et al. 2008). For this LDRD, our goal was to incorporate these two technologies into the Umbra environment and demonstrate its use in a proof of concept task. This chapter will describe the work put forth in connecting Umbra to these modeling technologies and using these technologies for creating realistic human behavior in the Sandia's Augmented Reality Training System (ARTS), an application built upon Umbra Simulation Framework (Gottleib et al. 2002).

2.2. Overview of ARTS

The ARTS platform uses augmented reality to create a more realistic training platform for protective force members who may engage in close quarters combat operations. Purely virtual training environments such as a PC game platform may assist in training tactics but do not allow for the integration of physical coordination between the members of the protective force team. Live training exercises may provide the accuracy for all facets of the task but are prohibitive due to the cost and the difficulty of conducting exercises in working facilities. ARTS provides the ability to train both tactics and physical coordination of teammates in a cheap, repeatable manner.

The ARTS platform relies upon using a room adorned with sensors that monitor the positions of light emitting diodes (LEDs) placed on humans or objects in the physical environment. ARTS uses Umbra to create and manage a virtual environment that incorporates representations of the tracked people and objects in the physical space. A trainee in the ARTS platform wears a helmet covered in an array of LEDs to monitor the position and orientation of the trainee as they move around the room. The helmet also has a camera that tracks the point of view from the trainee and a visual display that superimposes virtual artifacts from the simulation onto trainee's field of vision seen through the camera. The trainee also carries a life-like replica of a service rifle also covered with LEDs. A trainee's helmet and gun positions drive the position of a corresponding, armed virtual character in the Umbra simulation.

During the simulation, a trainee sees the world modeled in Umbra through their head-mounted display from the perspective of their virtual character. Each trainee controls their virtual character by moving around the room or using their weapon. Training scenarios in ARTS pit of a group of trainees against virtual adversaries. Trainees must maneuver in coordination with each other to quickly assess the situation and neutralize the threat.

To function as a team-training environment, the ARTS platform currently requires live trainees to be present to fulfill all of the team roles. While it would be possible to use hand-programmed virtual teammates to substitute for live team members, translating tactical knowledge into real-time, precise physical movements is a very difficult and time-consuming programming task. Using the programming by example approach, such as TAF, would serve as the better solution since it constructs a behavior model from observed physical movements. By observing several trials of a trainee, TAF will construct an accurate model where the necessary tactics emerge from the decisions made by the behavior model controlling a virtual character. Ideally, TAF should provide a model to drive a virtual teammate such that it performs task coordination and cooperation with human trainees in this environment just as well as a human counterpart.

Figure 1. Trainee performing in the ARTS platform.

2.3. Connecting TAF to Umbra

Accomplishing the goal of using TAF with trainees in the ARTS platform meant solving the problem of integration between the Umbra and TAF codebases. Both Umbra and TAF are developed by separate organizations using differing languages for constructing their code; Umbra is written in C++ while TAF and the Cognitive Foundry is written in Java. Also, there are serious constraints upon accurately representing the trainee perspective on the virtual world from Umbra into TAF. Previous applications of TAF looked at aerial combat situations or open terrain battlespace, both environments where the trainee can obtain perfect information about the location of adversaries. Umbra and the ARTS platform present a new challenge where objects can be occluded from the trainee's sight. For instance, an adversary can take cover behind an object in the virtual world. How does one know that the adversary is occluded enough such that the trainee does not detect them? If the trainee witnesses an adversary disappear from their line of sight, how does one represent the trainee's belief about the location of the occluded adversary?

Taking these challenges into account led to the development of a new Umbra package that allows for the data flow between the Umbra simulation and the TAF software. The figure shown below documents the series of modules required for connecting the representation of the trainee in the virtual Umbra world to generate and control a virtual teammate. These modules could be modified for learning the behavior for any entity in the Umbra simulation, but for now we limit the functionality to only the ARTS platform. The following description will start with the Umbra simulation represented in the top right corner and move counterclockwise to the TAF / Cognitive Foundry Connection module and finally to the virtual teammate avatar.

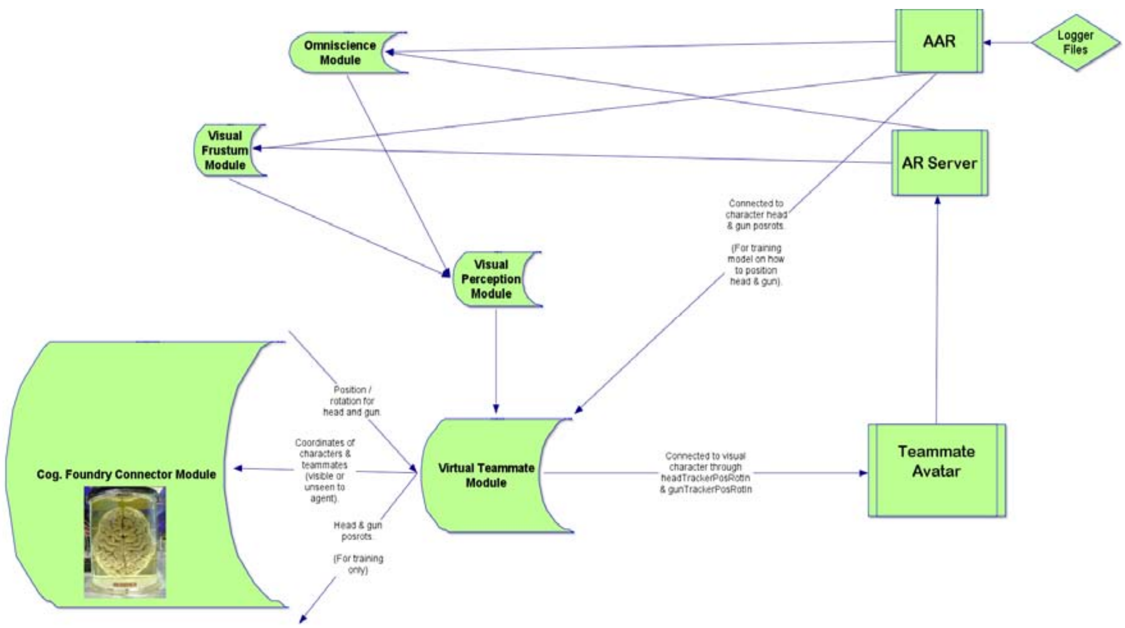

Figure 2. Data flow between ARTS platform and TAF

2.3.1. Data Source: Umbra Simulation

For collecting trainee behavior, the system must acquire information about the state of the world at a given time and the actions of the trainee (e.g. Where was the trainee located? Where was the trainee's gun pointed? Was it being fired?). The ARTS platform allows for constructing a live exercise or a replay of an exercise through its After Action Review (AAR) capability. In a live exercise, Umbra controls all simulated objects and people (e.g. adversaries) using instantiated models for each entity while trainees control their virtual representations in the motion-tracking room. In AAR, Umbra relies on logged data files that report the positions of each object and person, both simulated and actual, along with the person's actions at given times during the exercise. With the only difference between live exercises and AAR involving the controlling mechanisms for the objects and people in the virtual environment, we abstracted the data source and re-used the same modules for querying the simulation world for both live and replayed exercises.

2.3.2. Visual Perception Data

For the ARTS platform, a trainee relies only on visual stimuli for assessing their environment and making decisions needed to accomplish the given task. Decisions are based upon not only the immediately visible environment, but also predictions of the location of adversaries that are no longer in the trainee's line of sight. Since we don't have access to the trainee's state of mind as they perform in a live exercise for modeling purposes, one must manipulate the data from the simulation such that TAF can learn how a trainee handles an adversary once it leaves their field of vision. For completing this we combine data from two sensors to give this complete picture:

- Omniscience Module: This module reports the location and orientation of every object found within the virtual world. This data source is not sufficient for modeling since it provides perfect information on the world rather than the limited scope available to the trainee.

- Visual Frustum Module: Each virtual character in the ARTS platform, both simulated and human-controlled, has a visual frustum sensor attached at the head location. The sensor reports on the objects in the environment found within the frustum, thus approximating the trainee's line-of-sight in the simulation. The sensor also performs a polygon culling check to see if an object closer to the trainee's field of view occludes another object. If an object is flagged as occluded, the sensor will not list it as an object in trainee's line of sight. Though this better represents what the trainee views, it is also not sufficient for modeling since it excludes hidden objects from the data. That is, if an adversary moves out of a trainee's line of sight, the adversary's presence is still likely to have a significant impact on the trainee's behavior, but the model will be unable to represent this fact without any means to discriminate between an absent adversary and a hidden one.

The Omniscience and Visual Frustum modules feed into a Visual Perception module that joins these two data sets together. The output from this module will list the position and orientation of every object in the simulation along with a boolean flag reporting whether or not that object is visible to the trainee. From this data source, TAF receives a more accurate representation of the trainee's mental model formed on the virtual environment.

2.3.3. Virtual Teammate Module

The Virtual Teammate module serves as the bridge between the TAF model and trainee's virtual character. The Virtual Teammate module collects all information on the trainee's surroundings in the environment along with the actions taken by the trainee at that time. For ARTS, the observable actions from the motion-tracking system include the position and orientation of the trainee's head and weapon. The data sent from Virtual Teammate module to the TAF software include:

- Position and orientation for head on trainee's virtual character.
- Position and orientation for gun on trainee's virtual character.

- Output from Visual Perception module.
- Current time from Umbra simulation.

The Virtual Teammate module connects to the avatar representing a protective force virtual character. The TAF software, whether it is learning or executing a model of behavior, dictates how the Virtual Teammate module manipulates the avatar. If TAF is observing trainee behavior for learning a model, the Virtual Teammate provides the given action data directly as input to the avatar. If TAF is being used to control a virtual teammate, then TAF will provide the Virtual Teammate module with the new position and orientation for the virtual character's head and gun based upon the current input. The Virtual Teammate module updates the virtual teammate's avatar in the live exercise based upon the data.

2.3.4. Cognitive Foundry Connection Module

This module serves as the bridge between Umbra and the TAF code bases. The bridge is developed through a series of Java Native Interface (JNI) calls that allow Umbra to update the world representation for TAF and query TAF for its prediction on the next action for the object(s) of interest. The world representation for TAF involves the position and orientation of a target object(s), position and orientation of other objects in the world, and the current simulation time. The world representation is then transferred into the Cognitive Foundry where a model is either formed or queried based upon the current application. Since Umbra can easily provide the data required for TAF for any object, the Cognitive Foundry Connection Module is easily extensible for any Umbra simulation. As well, the world representation can also be utilized by AEMASE to allow for evaluation of a trainee based upon their current behavior. This module alone provides a major capability for combining the research between Umbra and Cognitive Foundry developers.

2.4. Demonstration of TAF in Umbra

As a proof of concept in using TAF in Umbra applications, we developed a demonstration using TAF with the ARTS platform to show the possibility in creating virtual teammates for training exercises. Working with ARTS developers, we obtained data logs from past demonstrations of the ARTS platform with different team sizes training for a variety of scenarios. To simplify our task for the demonstration, we selected a recorded scenario showing a single human trainee executing a room clearing exercise. In the exercise the trainee must locate and neutralize two adversaries in a room used for processing nuclear material. A chemical fume hood exists in the center of the room that occludes the adversaries from the trainee's field of vision once they enter the room. From observations of the recorded scenario, we witness the trainee entering the room and quickly moving toward the hood. The trainee moves around the left side of the hood to keep cover and position themselves to fire upon the adversaries. The trainee moves from cover and opens fire on the adversaries, neutralizing both in the firefight.

Our goal for TAF in this demonstration is to observe the trainee in the exercise and perform a rote memorization in order to reproduce the trainee's behavior in a live version of the exact scenario. For learning trainee behavior, we use the After action Review capabilities of the ARTS platform to replay the recorded scenario. In the After Action Review, both the adversaries and trainee's virtual character are controlled by time-stamped data covering the position, orientation, and action of each during the exercise. Prior to starting the replay, we instantiate the TAF/Umbra connection system modules defined in the earlier section and connect them to the trainee's virtual character. As the AAR progresses through the replay, TAF receives world information from the perspective of the trainee and forms the behavior memorization.

At the conclusion of the replay, we start the ARTS platform for a live exercise and load the scenario used in the recorded exercise. Instead of connecting the trainee's virtual character to a human working in the motion-tracked physical space, we connect TAF to the virtual character and instruct TAF to use the model just created for controlling the character. Once the exercise begins, we observe TAF moving the virtual character in a similar fashion as the trainee demonstrated in the recorded scenario. The TAF controlled character heads toward the fume hood, and moves around it to maintain low visibility to the adversary. Once in position, the character moves away from cover and into position to fire. In this demonstration, the TAF controlled character does not fire upon the adversaries due to inability on both the ARTS and TAF software for allowing the model to fire the weapon.

Despite this drawback, the demonstration successfully proved the ability to use TAF in the ARTS platform for creating accurate representations of human demonstrated behavior. After generating these behavior models, one can introduce them into a live exercise in the ARTS platform to have TAF substitute for a missing human trainee in a given scenario.

2.5. Conclusion / Future Work

This project completed the difficult task of integrating TAF and the Cognitive Foundry into Umbra. This major milestone allows Sandia researchers from the modeling and simulation and cognitive systems communities to work together on more applications focused on creating and utilizing generalized models of human behavior. We plan on studying two-man teams in the ARTS platform to create models from trainee behaviors that can generalize across novel situations. At the conclusion of the project, our goal is to create a virtual teammate in these types of scenarios that will provide a similar benefit to a trainee as that of a human counterpart.

3. SCALING UP TO INCREASINGLY RICH AND INTERACTIVE VIRTUAL CHARACTERS[1]

3.1. Introduction

The ability to simulate increasingly rich and interactive human characters is important for many possible national security applications, particularly for systems designed to train skills involving social engagement. The richness and complexity of human social behavior makes authoring the behaviors of such characters a great challenge. To support such efforts, this project has integrated Delta3D with the ABL (A Behavior Language) behavior engine (Mateas & Stern, A Behavior Language for Story-Based Believable Agents, 2002). ABL is a powerful reactive-planning language, providing a basis for authoring characters that simultaneously maintain a variety of goals and act to achieve them in an opportunistic manner.

While ABL provides great expressive power, authoring interactive characters that exhibit an increasingly rich set of possible behaviors presents a considerable organizational challenge. We turn here to the field of sociology to begin to get a handle on how to analyze, encode, and arrange a diverse set of interdependent social behaviors. To create more socially competent, believable, human-like autonomous agents, sociological tools can be leveraged to more closely model human behavior. In particular, there exist sociological tools that would aid in the creation of believable agents for use in interactive narratives.

Interactive narratives often include a small number of agents in a particular setting interacting with one another in social ways (Mateas & Stern, Structuring Content in the Facade Interactive Drama Architecture, 2005). A well suited tool for understanding and predicting behavior in this type of context the sociological notion of dramaturgical analysis (Goffman 1959). Dramaturgical analysis views social interactions in the metaphor of a drama; actors, roles, props, setting, audience, and stage are all identified. This metaphor is particularly useful when modeling self-presentation, or the manipulation of how one is perceived by others. When social interactions are seen through this metaphor, the reasons behind their behavior become more decipherable. Models of personality-specific social game behavior can then be constructed through dramaturgical analysis and can be used to inform the behavior of agents in interactive narratives.

In this chapter, we present the notion of a *social game*, a pattern of multi-agent interactions whose function is to modify the social state existing within and across the participants, which is derived from cases of applied dramaturgical analysis. Additionally, we provide an ontology for describing social games as well as a social-game-centric ontology for character personality. Lastly, we provide a framework aimed at producing more socially coordinated and believable interactions in autonomous agents by leveraging the structure of the dramaturgical metaphor.

[1] A substantial portion of the text in this chapter is copyright © 2009, Association for the Advancement of Artificial Intelligence, and is to appear as "The Computation of Self in Everyday Life: A Dramaturgical Approach for Socially Competent Agents" by Josh McCoy and Michael Mateas in the Proceedings of the *Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09).*

3.2. Related Work

Research on autonomous synthetic characters that behave in human-like ways has been an active research area in recent years (Loyall, 1997). Each vein of exploration in this research area has different foci and interests. The distinction we are interested in is that of social and non-social research.

Based on neuroscience and how the brain works mechanically, the ACT–R 5.0 Architecture (Anderson, 2004) is composed of perceptual-motor modules, a goal module, and a declarative memory module each of which are associated with specific cortical regions. Although this architecture does well in using neuroscience to inform a model of cognition, it has no explicit method of interacting socially with humans or other agents. STEVE, an embodied conversational agent, provides believable interaction with users in the setting of training users in problem solving on a naval ship (Rickel & Johnson, 1999). This interaction in this system is believable but is restrained to handle only interactions related to the group task and has no capability for more meaningful social interaction with the users.

Thespian (Si, Marsella, & Pynadath, 2005) and the system it is built on, PsychSim (Marsella, Pynadath, & Read, PsychSim: Agent-based Modeling of Social Interactions and Influence, 2004), are multi-agent capable systems that model social interactions based on models of social influence. Each agent has goals, actions it can perform, beliefs (including a recursive model of other agents), and mental models to increase the efficacy of the simulated agents' behaviors. The social interactions are based solely around the rules of social influence; more expressive social state behaviors on which to base social behavior would increase social believability.

Some research has focused on social group interactions. One example is the SGD Model (Synthetic Group Dynamics Model) of multi-agent social interaction which is based on having each group member being aware of the other group members and of the group itself (Prada and Paiva 2008). The SGD Model is based around four levels of agent knowledge: the individual, group, interactions, and context levels. We depart from this model by agents who are more motivated by social and emotional considerations and have social group interactions framed around a dramaturgical metaphor to take advantage of a wide array of sociological theory.

3.3. Dramaturgy and Social Games

One way to get a sense of the concepts needed to represent and make use of social games is to critically observe social interactions with the intention of noticing social games via a dramaturgical perspective. We have two needs to fulfill when choosing a source of social interactions to observe: (1) a dramatic setting (to better correspond to interactive dramas); (2) access to a rich and plentiful source of social interactions. To this end, we decided to study a television show with a focus on dramatic interactions and character-driven story: the HBO show *Sex and the City*.

In this study, the situations and actions taken by the characters were cast into a dramaturgical metaphor. Individual interactions were viewed in terms of the roles taken by the participants, the setting of the interaction, teams composed of the participants, who comprises the audience, etc. With the interactions represented in this dramaturgical way, social games became easier to distinguish from within the drama. This process of viewing and interpreting a social situation is known as dramaturgical analysis.

In order to turn observations into social games, criteria are needed to perform the screen action to social game mapping. The first criterion consists of identifying a set of screen actions that are directed toward a common set of social state changes. Secondly, the set of social state changes brought about by the screen actions must further the goals of some subset of participants in those actions. The application of these criteria to observations generates a set of social games, each of which contains at least one actor (participants in the performance who are not part of the audience) who has a goal consisting of the set of social state changes that result from the associated set of actions.

The dramaturgical analysis of *Sex and the City* yielded several attributes of social games. First, a group of social games observed were analyzed dramaturgically. They were then decomposed into a list of causally and temporally related events and a set of dramaturgical properties. From these analyzed social games, a schema for representing social games was developed.

The schema for representing social games consists of a description of the social game's dramaturgical qualities, the social and world state preconditions for the game to start, a dependency graph of social game events, and the state changes enacted by game completion (see figure 1). The dramaturgical qualities consist of a list of roles and their requirements, qualities needed in the setting, teams among the actors, and what qualifies as an audience for the social game. Social game events are composed of a list of participating actors, temporal properties, actions taken by actors, functional world change, and social facts modified by the event. Optionally, the events can reference other social games to create a hierarchical decomposition of social games. Because the social effects of social games can be different than the sum of the changes specified in its events, the state change upon the games' successful completion is also represented.

Other properties of social games that resulted from the dramaturgical analysis are that social games are hierarchical and sets of social games can be played in parallel. Furthermore, personalities and emotional states of the actors have a great deal of influence on the performance of the actions taken by the actors when performing in social games. To illustrate the results of the study, an example of a social interactions analyzed in the study is helpful.

To begin exploring a scene in *Sex and the City* in a dramaturgical metaphor, some of the more general dramaturgical attributes need to be related to the social situation. The setting of the performance is composed of a wedding engagement party held in an expensive apartment in Manhattan. The audience consists of several dozen upper-middle class married individuals who are all in some way socially connected to the newly-engaged couple. The props present are objects typically found at celebrations: champagne glasses, tables, chairs, presents, etc.

The cast consists of two single friends, Miranda and Carrie, and a group of several female acquaintances, all of whom are married. Carrie and Miranda have two very distinct personalities. Carrie is an outgoing person who tends to directly face situations one at a time and is very focused on the role she is playing. Miranda, while being focused like Carrie, prioritizes the avoidance of bad things over proactively seeking her goals.

The scene begins and plays out in the following sequence of events. Carrie, Miranda, and the group of women are engaging in conversation when the topic of relationships is brought up. The group of women discusses their current relationships. Eventually, the focus is placed on Carrie and she is asked about her relationship status. She states that she is single and content then passes the conversational back to the group. They then proceed to similarly ask Miranda about her relationship status. Miranda responds by going into a round of self deprecating jokes about her being single. She then excuses herself from the scene and exits the stage with Carrie; the scene ends.

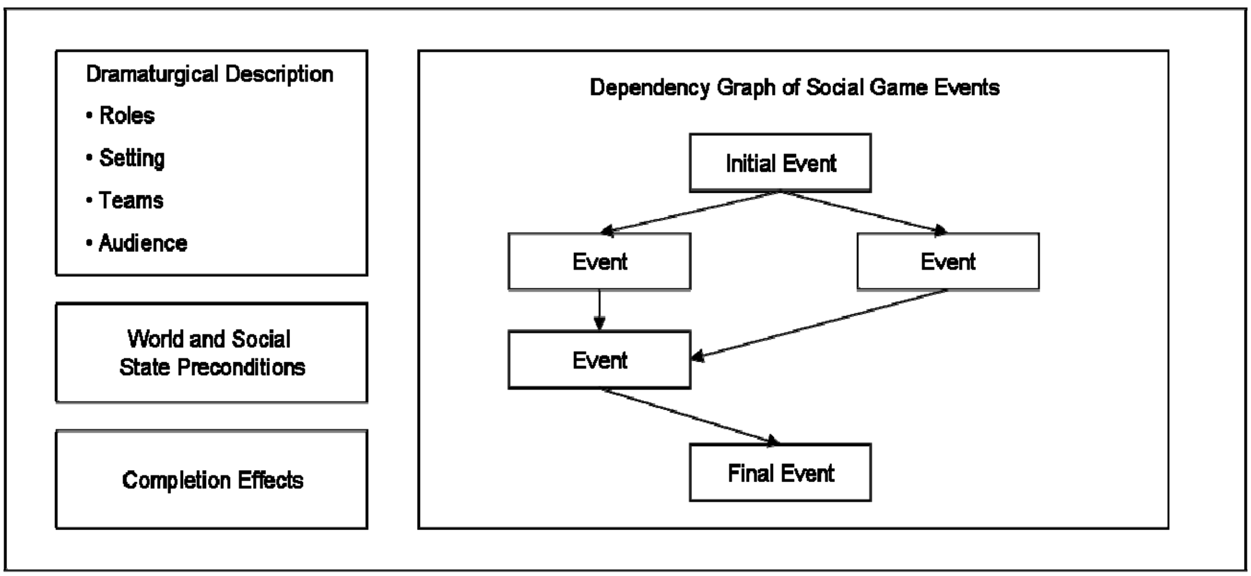

Figure 3. Overview of social game representation.

When this scene is considered with the criteria of determining social games from scene actions, a clear game is present in the previous example. The actions taken by Carrie, Miranda, and the group of women result in the social state change associated with what the group will think of those who do not share a vital part of the group's identity (the identity of being in a relationship). Two characters, Miranda and Carrie, have the goal of using the actions in the scene to manage the impression that the group of women will form of them.

This social game illustrates an important concept related to dramaturgical analysis: impression management. Impression management is a goal-directed attempt at influencing the perceptions that are formed by others, which can be performed either consciously or unconsciously. It is not necessary for the subject of the managed impression to be the one performing the managing; the impressions can be of another person, event, or arbitrary object. When one manages the impressions of oneself, it is called self-presentation. Impressions are managed by regulating social information and interactions. Both Miranda and Carrie had the goal of self-presentation with regards to the group of women in the social game.

Seen in terms of the social game representation schema, the preconditions of this game are that there is a group with a similar social status that another, smaller group does not possess. Each of the events in the scene has associated social change, temporal qualities, actions taken, functional change, and list of participating actors. Taking Carrie's response to questioning as an example, the social change is that Carrie establishes herself as an outsider of the group, while keeping the repercussions (such as stigmatization or being ostracized) to a minimum. Temporally, Carrie's response happens after the questioning by the group of women, before she relinquishes the conversational turn back to the women, and takes an amount of time associated with her discourse. Her actions consist of turn taking, a discourse act, and turn giving. The other participants perform the action of listening to Carrie. Movements involved in keeping conversational distance, lowered levels of champagne in glasses, and other common world changes associated with parties comprise the functional storyworld change. Finally, the participating actors are Carrie as the speaker and Miranda and the group of women as listeners. The dramaturgical qualities have been previously stated. Finally, the state change at the completion of the game is primarily comprised of the fact that the game was successfully completed and none of the actors broke the game structure or refused to play.

The context of the episode around the scene shows the compositional nature of social games. Because Carrie and Miranda are friends of the groom-to-be, they are playing a social game of supporting their friend's engagement. The example game played with the group of women was an event in the larger social game of supporting a friend. Furthermore, by planning to go to the party together, Carrie and Miranda are playing a social game of mutual support while simultaneously playing the example social game.

3. 4. Social Game Representation Areas

Employing dramaturgical analysis requires an ontology and representation for reasoning about social games. Unfortunately, such an ontology that supports the areas of representation needed does not exist. As part of our study of *Sex and the City*, basic knowledge about the areas of representation needed to represent and reason over social games began to coalesce. The most important parts of representing social games are the description of the social games and the description of the personalities of those playing the social games. These descriptions are important as the richness of the performance of social games is found in the variations of play, such as impression management and actor-specific event actions, in social games which are a product of the details of the actors' personalities and the qualities of the social game. The other areas of social game representation support those descriptions.

We present the following categories as part of the preliminary work done on an ontology used to represent the social state needed by our architecture.

Personality and Social Game Descriptions. Both social games and personalities are described in a way that our architecture can reason over their contents. The social game representation schema was previously explained in the Dramaturgy and Social Games section and can been seen in Figure 1.

Personality as used in this ontology is based on addressing the issues raised by studying social games. Primarily, the personality description needs to capture both the dimensions of variation in social games and the richness of individual impression management. This extends through the choosing of social games to initiate, negotiating what roles to play in social games with other participants, and the variation seen in the performances of the actions in social game events. Trait theory (McCrae & John, 1992) is often used in agents to describe personality. However, it lacks the expressiveness for detailed impression management. To address this requirement for expressiveness, there is a need for a personality description that allows for context-independent attributes, such as social game choosing parameters, and context dependent attributes like reactions to different classes of social games (such as reciprocity and affinity games) and storyworld state (like claustrophobia or sense of personal space).

Emotion. The role of emotions on behavior has been explored in depth (Ortony & Turner, 1990) and has been explored in agent architectures (Marsella & Gratch, EMA: A Computational Model of Appraisal Dynamics, 2006). However, the social role of emotions is the primary consideration in supporting the simulation of social games.

Social emotions (Parkinson, Fischer, & Manstead, 2004), or emotions intrinsically linked to social concerns and that cannot exist without a social component, are prominent in the motivation of characters that can play social games. The social emotions that are likely to be the most used in this ontology are the ones that have been the most often cited in literature and have associated appraisal models; jealousy, gloating/schadenfreude, guilt, gratitude, envy, anger/rage, and admiration are examples of social emotions that meet the criteria (Hareli & Parkinson, 2008).

By keeping an emotional state that includes social emotions, agents can evaluate the social state change of a social game and modify their emotions accordingly. Through the modification of emotional state after each social game, the emotionally influenced actions and decisions made while playing social games will vary more believably and richly when simulated.

Beliefs. The facts present in an agent's memory do not represent the ground truth of a world, as each agent's conception of the world is a production of its past experiences and current social state in keeping the concept of the semiotic self. In keeping with the concept that everything is social from the dramaturgical metaphor, all facts, including those that are mundane facts of the physical world, are stored in the agent's memory.

In order to categorize facts in a convenient way, inspiration was taken from Searle's ontology of fact representation (Searle, 1995). Facts are partitioned into basic facts, or facts that do not change with social influence (like the height of a mountain), and social facts that can change independently of the physical world.

Social facts comprise the agent's view of their social state. Status facts and institutional facts (socially inferred statuses and facts that cannot exist without a social context, respectively) as well as relationships can be represented with a slight variation of Searle's status facts notation. This variation is: X is associated with Y in context C where X and Y are world objects or compositional social facts and C is a context consisting of a set of objects or social facts. This representation is useful for representing concepts like social currency or reciprocity. For example Agent1 is associated with Agent2 in the context of ReciprocityDebt.

A special set of social facts are used to represent the concept of social norms (Goffman, Stigma: Notes on the Management of Spoiled Identity, 1963). Social norms are the set of social expectations that are a production of the culture the agent represents to be considered normal, or an individual who does not break the social norms. These norms are used in social game negotiation to help parameterize the dramaturgical qualities of the game. They also bring the potential for cultural conflicts to be present in social interactions.

3.5. Architecture Overview

To produce a system that simulates social games in a human-like way, there is a need for an architecture that is designed to handle the complexity of choosing social games and the flexibility to allow for the wide range of performance variation found in social games. The following is a preliminary work on an architecture design that would allow for the simulation of social games.

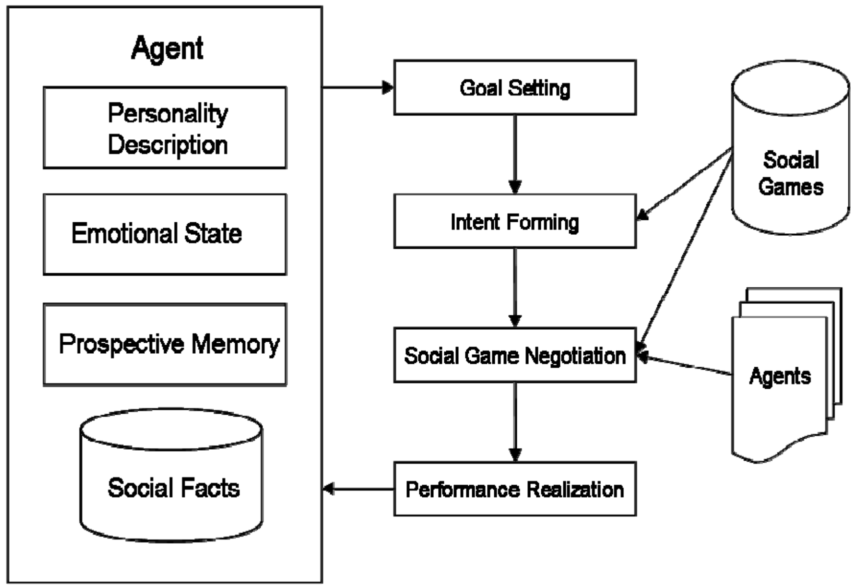

Figure 4. Overview of the autonomous agent architecture for playing social games.

The agent architecture is comprised of several components: the agent and its constituent pieces, the goal setting processes, the intent forming process, the game negotiator, and a database of annotated social games (an overview of this architecture can be seen in figure 2). This system is designed to model the process of human goal setting and turn the goal into an actualized social game using dramaturgical analysis as an organizational framework.

The goal setting process begins by employing several partial behavior theories developed in social science (some of which are described later in the Goal Setting section). An agent personality description is used to depict the types of goals the agent would be most likely to choose. A set of likely goals, such as to lower reciprocity (sociological concept of indebtedness) with a specific agent or to obtain power to influence another, is generated. Each goal is assessed for importance and persistence with respect to the agent (as described in Goal Volition section).

Next, the most important goals are matched with social games that will further their attainability. Other agents are notified that the agent wants to play a social game. This begins the game negotiation process where the details of the social game (role assignments, setting, audience, teams, etc) are determined. When the game details are successfully attributed, the game is enacted in the simulation world. The social, emotional, and physical ramifications of the game are then assessed by each agent. Each agent updates their emotional, social, and goal setting states according to the assessments. The cycle is then repeated starting with the top level of the goal setting process.

3.6. Goal Setting

In order believably to operationalize this notion of social games in an autonomous agent, the agents need a system of reasoning over social games and of choosing which social games to play according to their personality description and the current storyworld context. This necessitates instilling the ability to reason about and set goals to guide the choice of social games.

At the highest level of goal setting, the agent is informed by a more sophisticated version of the theory of ultimate psychological hedonism (UPH) (Mees & Schmitt, 2008). In a way similar to how the maximum expected utility function in a rational agent guides many existing agents toward actions that should attain maximum benefit, UPH lays the framework for social agents to further social goals. In its ancient, original form, UPH states that one approaches physical pleasure and avoids physical pain which provides a basic motivational factor for behavior. Modern adaptations have extended UPH to include emotion as well as physical pleasure and pain to be respectively approached and avoided.

While the modern version of UPH provides a general motivation for behavior, it is not rich enough to support the construction of an agent that performs goal setting in a human-like way. Variations in personality, such as emotional tendencies and senses of social norms, are not accounted for by UPH. Additional theoretical tools are required to build such an agent.

The following theories support goal setting in a human-like way. This list should not be considered complete or final because goal setting is not a solved problem. Each theoretical tool solves a small part of the goal setting problem and can be integrated with the other theories to provide a higher degree of human-like competency in choosing goals for social games.

TMMO. The modern notion of UPH is the basis for the two-dimensional model of metatelic orientation (TMMO). TMMO places individuals in a two-dimensional space with the first dimension being approach/avoidance. One dimensional extreme is the approach of pleasure while the other is the avoidance of pain. The second dimension is directness or indirectness by which the goal is explored.

TMMO also makes a distinction between content-based (telic) or emotional-based (metatelic) goal motivations. Content-based motivations consist of low social and emotional impact goals such as walking to the grocery store or playing the role of a cashier. Emotional-based motivations are much more involved with social and emotional situations. Gaining retribution from slander or positioning oneself for a job promotion are both goals that are emotional-based.

Hot and Cold Social Cognition. Social cognitive research provides further richness to goal setting through the concepts of role cognition and role focus (Lynch, 2007). Role cognition refers to the thought put into choosing and performing a role in a social game, while role focus is how cognitively and behaviorally consumed one is with the role.

The intensity of role cognition and role focus is denoted by a “hot/cold” metaphor. Hot role cognition refers to a high level of effort put into performing a role while cold denotes a blind following of a social template or schema when performing a role. Similarly, hot role focus is characterized by single-mindedly adhering to one role in one social game. Cold role focus means performing many roles serially or in parallel.

Social Emotions. As previously stated, social emotions are related to social games. In relation to social game choosing, the desired state of social emotions can be a pleasure to be approached, a pain to be avoided, or as something to manipulate in other participants. Furthermore, social emotions can be motivators for conscious impression management. To return to the *Sex and the City* example, Miranda was motivated by the social emotion embarrassment when she decided to use humor as a tool for self-presentation during the example social game

Goal Volition. An agent that plays social games needs to have a mechanism for comparing the importance of the current social game with social games that could be played.
When considering alternate social games to play, the agent runs the risk of seeming single-minded if one social game is doggedly perused. Alternatively, an agent who constantly and rapidly switches goals seems unintelligible. Furthermore, if goals are chosen as important to the agent and are subsequently forgotten because they were unobtainable when set, the agent could be seen as vacuous or lacking human conviction.

Action psychology introduced the concept of goal volition, or a measure of the persistence an agent has in goal pursuit (Dholakia & Bagozzi, 2002). The strength of volition for a goal is dependent on two major factors: goal intention and implementation intention. Goal intention is characterized by the desirability of the goal as determined by the agent, while implementation intention is a function of how well-formed the plan is to reach a goal. A goal that is desirable and is associated with a detailed plan for realization is more likely to stay an active goal with the agent than one that is not liked and vaguely planned.

Another important, related concept is that of prospective memory. Prospective memory is used to store goals that are not immediately obtainable but still have a high level of volition. Conditions to attain goals in prospective memory are acted upon when the volition of a stored goal is high enough to become an active goal. Additionally, active goals that can no longer be satisfied in the current world or social state are put into prospective memory to await the return of favorable conditions. Each goal in prospective memory is assessed for viability and has its associated volition updated during the goal setting process.

3.7. Intent Forming and Social Games

After a set of goals are established according to the agent’s personality description, they need to be refined into a set of actions and world state changes that can be manifested in the simulation. This process is known as forming intent. The practical implication of intent forming in this architecture is that it maps goals into social games an agent wishes to play. They resultant choices of social games are parameterized by aspects of the personality description. Additionally, the state of the storyworld has to fulfill the preconditions and the dramaturgical descriptions set by the chosen social games before the game can be a validly formed intent.

From the set that match the dramaturgical requirements and fulfill the social game preconditions, one social game needs be chosen to be sent to the other agents to start the role negotiation process. To make this decision, the agent must match its goals against the social game's completion effects and the changes caused by each individual event. If the social state changes of the social game match the goals of the agent, the social game's events are examined to determine if any of the personality description is violated. If there is no violation, the game is chosen and is sent to the role negotiation process.

3.8. Role Negotiation

After a social game has been chosen by an agent, the remaining details of the dramaturgical metaphor need to be determined. As the agent (who wants to play a particular social game) cannot assign roles to other agents only according to its interests, the potential participants need a chance to weigh in on their role in the proposed social game. Such is the responsibility of the role negotiation process. It is important to note that this process is typically one that happens on an extremely short time scale with humans; intent is formed, roles are taken, and social games are enacted many times in every conversation people participate in.

When an agent decides to play a social game, all of the potential participants are notified. Potential participants can be either those intended to have a role by the initiating agent, agents who wish to opportunistically take a role in the social game to attain their own goals, those who are in back stage teams, and those who are potential audience members. Agents can accept a social game role based on either their goals or their willingness to participate in social games that are goal neutral (as set in the agent's personality description).

The setting of the social game is largely a function of locations of the participating agents. If the initiating agent wants to better plan the location of the social game, other social games (such as asking another agent to move locations) can be used to create a deliberate setting. This type of deliberation is a direct result of hot role cognition and affects the volition of the appropriate goal by being adding to the implementation intention.

After the role negotiation process is successfully completed, the social game is ready for realization in the storyworld. However, the exact manner in which dialog and acts are involved in instantiating the social game has not yet been specified. A system capable of generating actions and dialog acts from higher level descriptions, such as the dynamic generation of discourse structures (Strong and Mateas 2008), would complete the architecture.

3.9. Conclusion

In this chapter, the concept of a social game as an organizing principle for defining meaningful social interaction between believable autonomous agents in a story setting has been introduced. An emphasis was placed on the large amount of variation found in the performance of social games that stems from impression management and variations in the personality of the actors. It was also shown the way in which dramaturgical analysis is a useful tool for both extracting social games from social interactions and describing them in a form that can be reused in other contexts.

Broad categories of an ontology that can be used to express the needs of simulating social games were presented. This ontology explores the areas of representation needed to describe social games and personalities in a way that allows for a complex, human-like performance of social games in a storyworld.

Also described was an agent architecture, similar in structure to a BDI (Bratman, 1987) system, with an emphasis on playing social games. Components of the architecture were depicted with an emphasis on how they contribute to playing social games.

In the future, the concepts presented in this paper are to be instantiated as a complete system. The areas of representation of social games will be further expanded upon to capture a larger variety of social games and variability in those games. Finally, we intend to enrich the library of social games through further dramaturgical analysis of social interactions.

4. REFERENCES

Abbott, R. G. (2006). *Automated Expert Modeling for Automated Student Evaluation.* Albuquerque: Sandia National Laboratories SAND2006-0291c.

Abbott, R. G., & Basilico, J. D. (2008). *Studying Teamwork with Automated Behavior Modeling.* Las Vegas: Applied Human Factors and Ergonomics International.

Anderson, J. (2004). An Integrated Theory of Mind. *Psychological Review , 111* (4), 1036-1060.

Basilico, J. D., Benz, Z. O., & Dixon, K. R. (2008). The Cognitive Foundry: A Flexible Platform for Intelligent Agent Modeling. *Proceedings of the 17th Conference on Behavior Representation in Modeling and Simulation.* Providence.

Bratman, M. (1987). *Intention, Plans, and Practical Reason.* Cambridge: Harvard University Press.

Dholakia, U., & Bagozzi, R. (2002). Mustering Motivation to Enact Decisions: How Decisions Process Chracteristics. *Journal of Behavioral Decision Making , 15* (3), 167-188.

Frijda, N. H. (1986). *The Emotions.* New York: Cambridge University Press.

Goffman, E. (1963). *Stigma: Notes on the Management of Spoiled Identity.* New York: Simon & Schuster Inc.

Goffman, E. (1959). *The Presentation of Self in Everyday Life.* Garden City: Doubleday.

Gottlieb, E. J., McDonald, F. J., Oppel, F. J., Rigdon, J. B., & Xavier, P. G. (2002). The Umbra Simulation Framework as Applied to Building HLA Federates. *Proceedings of the 2002 Winter Simulation Conference.* San Diego.

Hareli, S., & Parkinson, B. (2008). What's Social About Social Emotions? *Journal for the Theory of Social Behaviour , 38* (2), 131-156.

Izard, C. (1977). *Human Emotions.* New York: Plenum Press.

Loyall, B. (1997). Believable Agents: Building Interactive Personality. *Ph.D. diss.* Computer Science Department: Carnegie Mellon University.

Lynch, K. (2007). Modeling Role Enactment: Linking Role Theory and Social Cognition. *Journal for the Theory of Social Behaviour , 37* (4), 379.

Marsella, S., & Gratch, J. (2006). EMA: A Computational Model of Appraisal Dynamics. *Proceedings of Agen Construction and Emotions.* Vienna, Austria.

Marsella, S., Pynadath, D., & Read, S. (2004). PsychSim: Agent-based Modeling of Social Interactions and Influence. *Proceedings of the International Conference on Cognitive Modeling*, (pp. 243-248).

Mateas, M., & Stern, A. (2005). Structuring Content in the Facade Interactive Drama Architecture. *Proceedings of Artificial Intelligence and Interactive Digital Entertainment* . Marina Del Rey, California.

McCrae, R. R., & John, O. P. (1992). An Introduction to the Five-Factor Model and its Applications. *Journal of Personality , 60*, 175-215.

Mees, U., & Schmitt, A. (2008). Goals of Action and Emotional Reasons for Action. A Modern Version of the Theory of Ultimate Psychological Hedonism. *Journal for the Theory of Social Behaviour , 38* (2), 157-178.

Ortony, A., & Turner, T. J. (1990). What's Basic About Basic Emotions? *Psychological Review , 97*, 315-331.

Parkinson, B., Fischer, A., & Manstead, A. (2004). *Emotion in Social Relations: Cultural Group, and Interpersonal Processes.* Psychology Press.

Parrott, W. (2001). *Emotions in Social Psychology.* Philadelphia: Psychology Press.

Prada, R., & Paiva, A. (2008). Teaming Up Humans with Autonomous Synthetic Characters. *AI Commun , 21* (1), 83-85.

Rickel, J., & Johnson, W. L. (1999). Virtual Humans for Team Training in Virtual Reality. *Proceedings of the Ninth International Conference on AI in Education*, (pp. 578-585).

Searle, J. (1995). *The Construction of Social Reality.* N.Y.: Free Press.

Si, M., Marsella, S., & Pynadath, D. (2005). Thespian Using Multi-Agen Fitting to Craft Interactive Drama. *Proceedings of AAMAS '05*, (pp. 21-28).

Strong, C., & Mateas, M. (2008). Talking with NPCs: Towards dynamic generation of discourse structures. *Proceedings of the 4th Artificial Intelligence and Interactive Digital Entertainment Conference.* Palo Alto: AIIDE.

DISTRIBUTION

1	MS0123	D. Chavez, LDRD Office	1011
1	MS0899	Technical Library	9536

Sandia National Laboratories