The UTEP AGENT System

David Novick, Iván Gris, Diego A. Rivera, Adriana Camacho, Alex Rayon, Mario Gutierrez
The University of Texas at El Paso
500 West University Ave.
El Paso, TX 79968 USA
(+1) 915-747-6031
novick@utep.edu, ivangris4@gmail.com,
{darivera2, accamacho2, amrayon2, mgutierrez19}@miners.utep.edu

Categories and Subject Descriptors
H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *artificial, augmented, and virtual realities.*

General Terms
Design, Human Factors

Keywords
Embodied Conversational Agents

1. INTRODUCTION
This paper describes a system for embodied conversational agents (ECAs) developed at the University of Texas at El Paso by the Advanced aGent ENgagement Team (AGENT) and one of the applications—Survival on Jungle Island—built with this system. In the Jungle application, the ECA and a human interact with speech and gesture for approximately 40 – 60 minutes in a game composed of 23 scenes. Each scene comprises a collection of speech input, speech output, gesture input, gesture output, scenery, triggers, and decision points.

The UTEP AGENT system, and the applications such as the Jungle game developed with the system, were created to enable research into rapport between ECA and human, with particular emphasis on the effects of paralinguistic behaviors on rapport and on the development of rapport over time. For example, the most recent study conducted using the Jungle game examined whether rapport would increase if the agent asked the human to perform task-related gestures and then perceived these gestures. A current study is examining the effect on rapport of differences in the ECA's speech between extraverted and introverted language.

In developing our system, full automation of the agent was a key consideration. To conduct our research on human-ECA rapport, we needed agents and systems capable of maintaining high levels of engagement with humans, and in some cases over multiple interaction sessions. These sessions can potentially extend to longer periods of time to examine long-term effects of the ECA's behaviors. Indeed, our design had to accommodate the possibility for long-term human-agent interactions extending over several sessions across several weeks. This would have been cumbersome and inefficient if done as Wizard-of-Oz system. Instead, we developed middleware that enabled us to create ECAs through a declarative approach, with XML-style scripting.

ICMI 2015, November 9–13, 2015, Seattle, WA, USA.
ACM 978-1-4503-3912-4/15/11.
DOI: http://dx.doi.org/10.1145/2818346.2823302.

2. SYSTEM ARCHITECTURE
The system's physical implementation uses Unity 4, a Microsoft Kinect, and the Windows Speech SDK, interfaced and networked with each other and synchronized to handle the agent's complex behavior. The system uses the Unity 4 game engine to display the ECA and Unity's Mecanim system to create an extensive array of animations.

The ECAs are built using a three-tiered architecture. At the bottom layer is the Kinect™ sensor, which supports RGB video, audio provided by an array of microphones, and a depth field based on infrared sensor information. The middle layer contains all the logic of the agent's behavior and sensor interpretation, including speech recognition, text-to-speech or audio playback, and gesture recognition. The top layer contains scripts in Unity3D Game Engine, which renders and animates the ECAs and contains the virtual environment (scenery) in which they appear.

The system's software implementation includes over 20 modules, presented in Figure 1. In this paper, we focus on the three most important modules: animations, markup language, and gesture recognition.

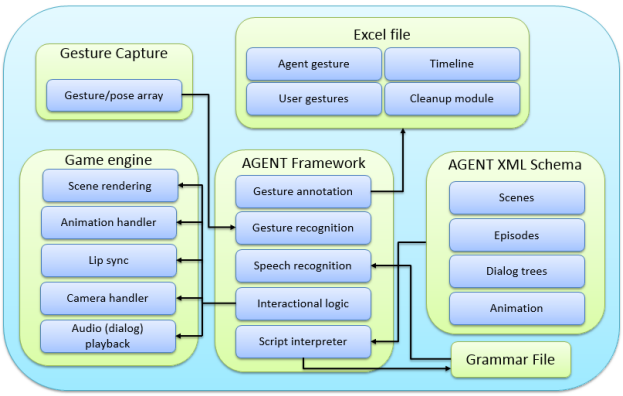

Fig. 1. AGENT system software modules.

2.1 Animations
Animations are played by a state graph that follows author-specified parameters of when an animation should start, end, or blend with another animation. Multiple animations can be blended to obtain a completely different animation in real time and to give the player the impression that the agent never moves in exactly the same way twice. Animations are divided layers that can control different parts of the body, so multiple animations can be played at the same time

and affect different limbs of the agent - for example simultaneously playing a blinking and talking animation on the face only, an explaining animation on the arms, and a walking animation from the hips down. The animation system is described in detail in [1].

2.2 Markup Language

To handle long-term interactions we developed middleware that parses, interprets, and executes XML files that define the scenes [2]. Each virtual environment where the agent can appear is known as a scene. Each scene can contain one or more episodes, which can branch depending on gesture or speech input. Each episode also contains recorded or synthesized dialogs and can contain grammar elements for speech recognition.

2.3 Gesture Recognition

We built a gesture tool using Microsoft's Kinect sensor. The tool itself is built as a standalone Windows application that can be connected to Unity3D, a game engine. In addition, the tool works both, as a way to recognize gestures, or to create gesture libraries that will later be used in the recognition [3]. We also use this tool to automatically log player's behaviors, potentially saving hours of manual coding, and to have our ECAs react to the player's movement in real time, sacrificing accuracy for speed.

3. SURVIVAL AGENT

The latest version of the agent, Adriana, leads the player through a series of activities and conversations while playing a game called Survival on Jungle Island. The system simulates a survival scenario where the player has to collaborate, cooperate, and build a relationship with the ECA to survive. In building the game, we sought to maximize rapport-building opportunities as well as to take advantage of the non-verbal behaviors in a more immersive environment, where both the player and the agent can interact with the same objects in virtual space. The storyline provides the flexibility and decision making without creating a completely open environment where tasks would otherwise be difficult to set up and evaluate.

The scenario comprises 23 scenes plus an introduction, where each scene lasts approximately five to eight minutes depending on the player's choices and interaction speed. Each scene was carefully scripted to enhance an aspect of a rapport-building interaction. The script provided a guideline for the relationship to evolve. The participants covered all scenes with Adriana with an average of 40 minutes of interaction. Although there was an implied one-hour limit to finish the interaction (the sign-in sheets had one hour slots), participants were told to take as much time as necessary. These times ranged between 30 minutes and 55 minutes.

The game starts with a cinematic sequence, which is a non-interactive video that offers some background for the situation. It explains how your ship is sinking in a storm. The few sailors were washed off the deck by a giant wave. The captain is unwilling to abandon his ship but sends you away with the hope that you might survive. This is followed by the first two scenes, which are meant to be instructive for the players without being an obtrusive tutorial that breaks the impression of an immersive reality. During the first scene, the player wakes up on a beach with someone—the agent—looking at the player. The agent introduces herself and asks a few questions, including if the participant is fine, if they can walk, and if they think there are other survivors. These questions help the player get immersed in the interaction. The agent then introduces herself and asks for the player's name. Most open-ended questions are not handled in the system by a language model. Rather, they are treated as wildcards, and the interaction is designed to traverse a dialog path through which any answer to those questions would seem recognized and valid (e.g., "Where are my manners? My name is Adriana. What is your name?" After the player responds, the name is not purposely recognized, and the ECA checks merely for an audio input, to which the agent always replies "Nice to meet you" and then continues the conversation.). These personal questions give the player an impression of seamless recognition and are meant indicate to the player that the agent is processing what it hears. This approach helps avoid command-like answers to questions, where players reply only with single words.

Fig. 2. Survival on Jungle Island.

The purpose of the second scene is to progress the story by providing a plausible explanation why there is a person (the ECA) in the jungle and how she survived thus far. This scene does react appropriately to a player's verbal response. At some point close towards the end of the scene, the ECA asks if they should stay on the same shelter or find a better place. Depending on the player's choice, the story develops.

About half the scenes have versions in which the agent recognizes task-oriented gestures by the player, such as spear-fishing or doing a "high five." In the final scene, the player and the agent are rescued by a helicopter.

4. ACKNOWLEDGMENTS

The authors would like to acknowledge Guillaume Adoneth, David Manuel, Joel Quintana, Anuar Jauregui, Tim Gonzales, Alfonso Peralta, Victoria Bravo, Brynne Blaugrund, Laura Rodriguez, Jaqueline Brixey, Yahaira Reyes, Paola Gallardo, and Nick Farber.

5. REFERENCES

[1] Gris, I., Rivera, D.A., and Novick, D. Animation guidelines for believable embodied conversational agent gestures. In *HCI International 2015*, Los Angeles, CA, August 2015, LNCS 9179, pp 197-205.

[2] Novick, D., Gutierrez, M. Gris, I, and Rivera, D.A. A Mark-Up Language and interpreter for interactive scenes for embodied conversational agents. In *HCI International 2015*, Los Angeles, CA, August 2015, LNCS 9179, pp. 206-215.

[3] Gris, I., Camacho, A., and Novick, D. Full-body gesture recognition for embodied conversational agents: The UTEP AGENT gesture tool. In *Gesture and Speech in Interaction (GESPIN 2015),* Nantes, France, in press.