SCREAM: Scripting Emotion-based Agent Minds

Helmut Prendinger
Dept. of Information and Communication Eng.
Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
+81 3 5841 6755
helmut@miv.t.u-tokyo.ac.jp

Mitsuru Ishizuka
Dept. of Information and Communication Eng.
Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
+81 3 5841 6347
ishizuka@miv.t.u-tokyo.ac.jp

ABSTRACT

SCREAM is a scripting tool that enables authors to generate emotionally and socially appropriate responses of animated agents. System users may design the mental make-up of an agent by declaring a variety of parameters and behaviors relevant to affective communication and obtain quantified affective reactions that can be input to an animation engine. While the default operations of an agent's 'mind' are based on psychological and sociological research, authors may easily modify and extend its rule set. In order to facilitate high-level scripting and connectivity with other web-based animated agent systems, our tool is written in a lightweight Java based Prolog system and Java.

Categories and Subject Descriptors

H.5.1 [**Information Systems**]: Multimedia Information Systems; H.5.2 [**Information Systems**]: User Interfaces

General Terms

Design, languages, theory

Keywords

Conversational agents, emotion and personality, human-like and believable qualities, scripting agents, agent modeling

1. INTRODUCTION AND MOTIVATION

Animated characters with interactive affective behavior have the potential to be beneficial for a wide variety of tasks, ranging from tutoring and product presentation to interactive entertainment [2]. In many cases, however, the success of those systems relies on the careful crafting of their designers, who are typically programmers. We believe that the growing popularity of animated agent systems will increase the demand for tools that allow content experts rather than programmers to script the behavior of agents in a simple and intuitive way. The SCREAM (**SCR**ipting **E**motion-based **A**gent **M**inds) system is such a scripting tool.

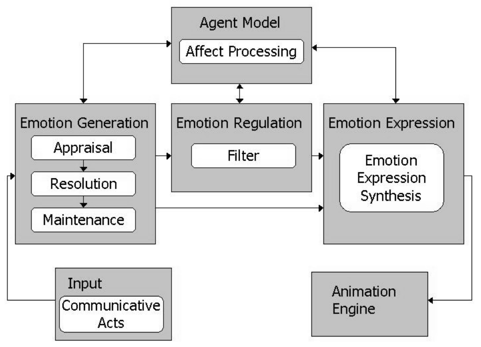


Figure 1: SCREAM System Architecture.

SCREAM is intended as a plug-in to task specific agent systems such as interactive tutoring or entertainment systems that provide possible verbal utterances of an agent. Our system may then decide on the kind of emotion expression and its intensity, based on a multitude of parameters that are relevant to the current interaction situation. Parameters are derived from the agent's mental state as well as the peculiarities of the social setting in which the interaction takes place and features of the agent's interlocutor(s), e.g., the user. Agents are adaptive in the sense that affective features of the interaction history result in updated values for certain mental states, such as attitudes and social relations.

As a scripting tool, the SCREAM system can easily be extended by adding or modifying rules that encode the agent's cognitive processes such as appraisal, or intensity combination functions. An important feature of our system is the *granularity* of agent behavior scripting. An agent's mental state can be designed at many levels of detail, from driven purely by (personality) traits to full awareness of the social interaction situation including agent-specific beliefs and beliefs attributed to interacting agents. The flexibility of our approach to scripting agent minds allows to create agent personalities with varying degrees of social sophistication.

2. SYSTEM ARCHITECTURE

The SCREAM system provides scripting tools that allow authors to control interactive emotional reactions of multiple characters in a natural way. A Java based Prolog system [1] is used to support high-level scripting of an agent's mind components: emotion generation, emotion regulation, emotion expression, and the agent model. An overview of the system architecture is given in Fig. 1. The SCREAM system also features an interface with an XML-style language called MPML (**M**ultimodal **P**resentation **M**arkup **L**anguage) [4] that supports easy control and synchronization of multiple agents' embodied behavior and synthetic speech.

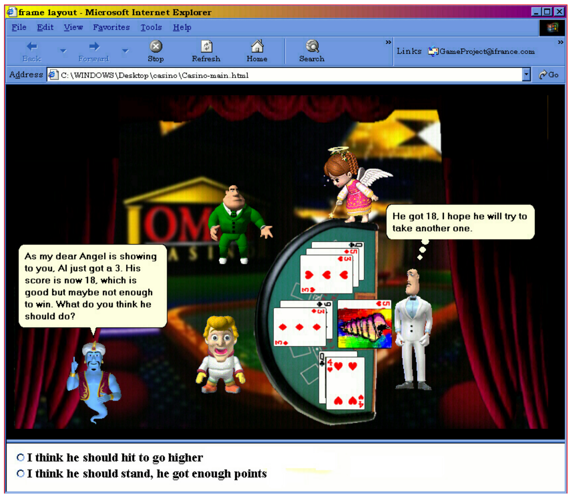
2.1 Emotion Processing

The agent receives input in the form of communicative acts that contain information about the conveyed meaning and modalities of the utterance. The *Emotion Generation* component contains modules for the generation and management of emotions. The appraisal process evaluates events in terms of their emotional significance for the agent, depending on its goals, standards, and attitudes [5, 4]. Since a multitude of emotions is often elicited at the same time, 'conflicting' emotions have to be resolved in order to obtain a unique candidate for emotion expression. The emotion resolution module filters emotions by considering their intensities and the agent's personality traits. The emotion maintenance module handles the decay process of emotions.

Communication is always embedded into a social context where participants take social roles with associated communicative conventions. The *Emotion Regulation* component contains a set of parameters that modulate the agent's expression of its emotional state, such as social variables (social power and familiarity), the agent's personality, as well as the interlocutor's personality and linguistic style. The resulting emotion with an intensity calculated from the described parameters is passed to the *Emotion Expression* component that directly communicates with the *Animation Engine* where it is instantiated to specific agent behavior.

2.2 Dynamics of Affective States

The *Agent Model* component contains the agent's character profile, including its personality traits, attitudes, goals, and beliefs. While intensity values of some features (e.g., traits and standards) remain invariant, most of them change during social interaction. We are primarily concerned with the dynamics of attitudes as a result of an agent's 'affective interaction history' with other agents. The psychological concept of *(signed) summary record* is employed to capture an agent's liking or disliking of its interlocutor [4]. In short, if some interlocutor triggers mostly positive (negative) emotions in the agent, it might change its attitude toward the interlocutor and be biased to appraise the interlocutor's future actions in a more positive (negative) way.


frame layout - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search
Links GameProject@ifrance.com
Address C:\WINDOWS\Desktop\casino\Casino-main.html
Go
OM
ASIN

As my dear Angel is showing to you, Al just got a 3. His score is now 18, which is good but maybe not enough to win. What do you think he should do?

He got 18, I hope he will try to take another one.

○ I think he should hit to go higher
○ I think he should stand, he got enough points

Figure 2: Casino Scenario.

3. ILLUSTRATIVE EXAMPLE

Our web-based interaction setting implements a casino scenario where a user can play the Black Jack game under the guidance of a virtual advisor. Fig. 2 depicts a situation where the character "Genie" (bottom left) practices Black Jack with the user by commenting the game of another character, "Al". All characters in the scenario utilize the Microsoft Agent package [3] as an Animation Engine.

Our intention here is to demonstrate how Genie's mental make-up as well as the (affective) interaction history with the user determine his behavior. Among others, we set Genie's character profile to agreeable and a initially positive attitude toward the user. For expository reasons, we let the user never follow Genie's advice. At first, Genie feels distress or sorry when the user looses. However, as the user repeatedly ignores the advisor's suggestion to either "hit" or "stand", Genie eventually gloats about the user loosing a game, and even resents the user's occasional good luck. Those negative emotional reactions are a consequence of Genie's increasing dislike of the ignorant user, which elicit emotions such as 'gloat' rather than 'sorry for' (the user). In accordance with his agreeableness, however, Genie expresses his negative emotions with low intensity.

On the other hand, when the user mostly follows Genie's advice, he will be sorry for the user upon a lost game (or ashamed when having given wrong advice), and happy for the user in case of a won game, as his attitude toward the user is positive. Since Genie is assumed as agreeable, he expresses positive emotions with high intensity.

The casino scenario is scripted at an intermediate level of granularity. Genie's goals, beliefs, and (initial) attitude toward the user are explicitly added to his character profile as facts, together with his personality traits and social relations. Events such as the user's actions and linguistic style are encoded in communicative act structures. A major concern about the SCREAM system is that it assumes a rich repertoire of 'canned' verbal responses that reflect both a specific emotion and its intensity. This problem can be alleviated by a lower level of granularity or obvious abstractions of affective reactions, e.g., into positive and negative ones.

4. ACKNOWLEDGMENTS

This research is supported by the Research Grant (1999-2003) for the Future Program ("Mirai Kaitaku") from the Japanese Society for the Promotion of Science (JSPS).

5. REFERENCES

[1] BinNet Corp. *Jinni 2000: A high performance Java based Prolog for agent scripting, client-server and internet programming*, 2000. URL: `www.binnetcorp.com`.

[2] J. Cassell, S. Prevost, J. Sullivan, and E. Churchill. *Embodied Conversational Agents*. The MIT Press, 2000.

[3] Microsoft. *Developing for Microsoft Agent*. Microsoft Press, 1998.

[4] H. Prendinger, S. Descamps, and M. Ishizuka. Scripting affective communication with life-like characters in web-based interaction systems. *Applied Artificial Intelligence Journal*, 2002. To appear.

[5] W. S. N. Reilly. *Believable Social and Emotional Agents*. PhD thesis, Carnegie Mellon University, 1996. CMU-CS-96-138.