Scripting the Bodies and Minds of Life-like Characters

Helmut Prendinger Sylvain Descamps Mitsuru Ishizuka

Department of Information and Communication Engineering
Graduate School of Information Science and Technology
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
E-mail: {helmut,descamps,ishizuka}@miv.t.u-tokyo.ac.jp

Abstract. In this paper, two systems will be described. First, we present an architecture for emotion-based agents, called SCREAM, that allows to encode affect-related processes for an animate character. Content authors may design the mental make-up of the agent by declaring a variety of parameters relevant to affective communication and obtain quantified emotional reactions. Second, we report on MPML, an XML-style markup language that facilitates the control and coordination of animated characters in web-based environments. Both systems are integrated such that the 'bodies' and 'minds' of life-like characters can be easily controlled.

1 Introduction

Artificial Intelligence (AI) is traditionally concerned with agents' intellectual skills that can optimize their efficiency or accuracy in completing certain tasks, such as planning, learning, or natural language understanding [17]. On the other hand, Hayes-Roth and Doyle's [7] work on 'animate characters' aims to make agents *life-like* or *believable* as well as more 'broad' in their competence, rather than efficient or accurate. Research in this direction has attracted significant interest recently, and life-like agents have already been developed for a wide variety of tasks, including tutor agents in interactive learning environments [9], presenter agents on the web [1, 8], and virtual actors for entertainment [16].

However, the success of many of those systems relies on the expertise 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 interactive behavior.

In this paper, we will describe two tools that may significantly facilitate the design of life-like characters: SCREAM and MPML. While the SCREAM system is intended to *animate* an agent, e.g., by giving it goals and attitudes (an individual persona), the MPML tool allows to control the agent's visual appearance as an *animated* character. We take 'life-likeness' as an umbrella term for agents that are both animate and animated.

SCREAM (**SCR**ipting **E**motion-based **A**gent **M**inds) is a system for scripting a character's 'mind'. The system allows to specify a character's mental make-up and endow it with emotion and personality which are considered as key features for the life-likeness of characters. A character's mental state can be scripted at many levels of detail, from driven purely by (personality) traits to having full awareness of the social interaction situation, including character-specific beliefs and beliefs attributed to interacting characters or even the user. MPML (**M**ultimodal **P**resentation **M**arkup **L**anguage) is a system that is responsible for scripting a character's 'body'. It facilitates the control and synchronization of the embodied behavior of characters.

The rest of the paper is organized as follows. The next section provides a step-by-step introduction to the core components of the SCREAM system architecture. Each of the modules is explained in terms of its role in the generation of an agent's affective behavior, together with details about its implementation. Section 3 briefly reports on MPML, a markup language for character control, by describing some of its tagging schemes. Section 4 demonstrates how our system works. In Section 5, we conclude the paper.

2 The SCREAM System

The SCREAM system allows authors to control interactive emotional reactions of multiple characters in a natural way. While the system is written in Java for portability, a Java based Prolog system called Jinni [2] is used to support high-level scripting of an agent's mind components: Emotion Generation, Emotion Regulation, Emotion Expression, and the Agent Model. SCREAM can be easily extended by adding or modifying rules that encode the character's cognitive processes. An overview of the system architecture is given in Fig. 1. Each of its components will be discussed in the following sections.

2.1 Emotion Generation

A core activity of an emotion-based agent mind is the generation and management of emotions, which is dealt with by three modules, the *appraisal* module, the *emotion resolution* module, and the *emotion maintenance* module. They will be described in the following. We start with a brief description of the input to the emotion generation component.

Input to an Agent's Mind. Input consists of communicative acts of the form

```
com_act(S,H,Concept,Modalities,Sit)
```

where S is the speaker, H the addressee, *Concept* the information conveyed by S to H in situation *Sit*, and *Modalities* is the set of communicative channels used by S, such as specific facial displays, acoustical correlates of (expressed) emotions, linguistic style, gestures, and posture.

Appraisal Module. Reasoning about emotion models an agent's *appraisal process*, where events are evaluated as to their emotional significance for the agent

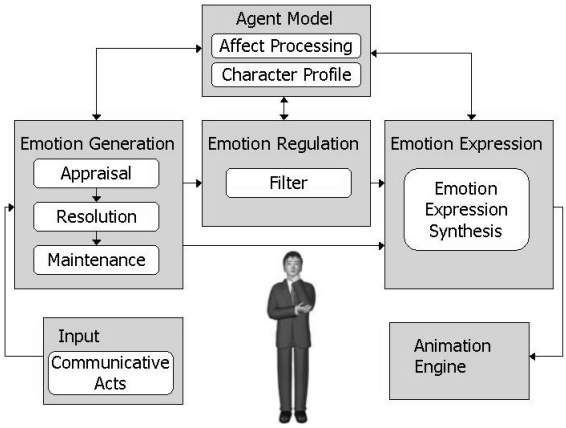

Fig. 1. SCREAM System Architecture.

[13]. The significance is determined by so-called 'emotion-eliciting conditions', which comprise an agent's relation to four types of abstract mental concepts: (i) *beliefs*, i.e., state of affairs that the agent has evidence to hold in the (virtual) world; (ii) *goals*, i.e., states of affairs that are (un)desirable for the agent, what the agent wants (does not want) to obtain; (iii) *standards*, i.e., the agent's beliefs about what ought (not) to be the case, events the agent considers as praiseworthy or blameworthy; and (iv) *attitudes*, i.e., the agent's dispositions to like or dislike other agents or objects, what the agent considers (not) appealing.

Following the emotion model of Ortony, Clore, and Collins [13] (the OCC model), we conceive emotion types as classes of eliciting conditions, each of which is labelled with an emotion word of phrase. In total, twenty-two classes of eliciting conditions are identified: *joy*, *distress*, *happy for*, *sorry for*, *resent*, *angry at*, and so on. Consider the emotion specification for fortunes-of-others emotion *resent* (being distressed about another agent's joy). The following rule is written in Prolog-style form close to the actual code:

$$\mathsf{resent(L1,L2,F,}\delta\mathsf{,Sit)}\ \textbf{if}\ \mathsf{directed\_to(L1,L2,Sit)}\ \textbf{and}\ \mathsf{dislikes(L1,L2,}\delta_{NApp(F)}\mathsf{,Sit)}\ \textbf{and}\ \mathsf{joy(L2,L1,F,}\delta_{Des(F)}\mathsf{,Sit)}$$

The rule reads as follows. The (locutor-)agent $L1$ resents agent $L2$ about state of affairs F in situation Sit with intensity degree δ if $L2$ is the addressee in Sit, $L1$ dislikes $L2$ with 'non-appealingness' degree δ_{NApp}, and believes that $L2$ is joyful about F with 'desirability' degree δ_{Des}. Whether this belief is true or not is entirely in the content author's control, and typically specified in the communicative act description. We assume intensities $\delta_i \in \{0, \ldots, 5\}$ such that zero is the lower threshold, i.e., the mental concept is not active, and five is the maximum value. By default, intensities δ_i are combined to an overall intensity δ by logarithmic combination $\delta = \log_2\left(\sum_i 2^{\delta_i}\right)$. Although this way to combine intensities seems plausible, content authors might wish to employ a different combination rule (e.g., additive), and edit the combination rule in question.

Since a reasonably interesting agent will have a multitude of mental states (beliefs, goals, attitudes, and so on), more than one emotion is typically triggered when the agent interacts with another agent. However, since an agent should clearly express a specific emotion at any given time, we need some way to resolve the agent's emotions. This problem will be discussed in the next paragraph.

Emotion Resolution Module. The emotions generated in an agent at a given time are called *active* emotions (in Sit) and are collected together with their intensities in a set $\{\langle E_1, \delta_1, Sit\rangle, \ldots, \langle E_n, \delta_n, Sit\rangle\}$. The presence of multiple emotions is resolved by computing and comparing two states. The *dominant emotion* is simply the emotion with the highest intensity value (the case where no unique dominant emotion exists will be decided by the agent's personality, see below). On the other hand, the *dominant mood* is calculated by considering all active emotions. Similar to Ortony [12], we distinguish between 'positive' and 'negative' emotions. Examples of positive emotions are 'joy', 'happy for', and 'sorry for', whereas 'resent' and 'angry at' are negative emotions. Then the dominant mood results by comparing the overall intensity value associated with the positive and negative emotion sets, which is obtained by logarithmic combination. The *winning emotional state* is decided by comparing the intensities for dominant emotion and dominant mood. Thereby, we can account for situations where an agent has a joyful experience, but is still more influenced by its overall negative emotions (mood) for another agent. In situations where equal intensities (of active emotions, moods, etc.) result, we consider the agreeableness dimension of an agent's personality. The agreeableness dimension is numerically quantified, with a value $\gamma_A \in \{-5, \ldots, 5\}$. Consequently, an agent with disagreeable personality (e.g., $\gamma_A = -3$) would favor a winning negative emotional state to a (winning) positive emotion if both have the same intensity level.

Emotion Maintenance Module. This module handles the decay process of emotions. Depending on their type and intensity, emotions may remain active in the agent's memory for a certain time during the interaction [15]. A decay function decreases the intensity levels of the active emotions each 'beat' by n levels until the intensity is equal of smaller than zero. A beat is defined as a single action-reaction pair between two agents. The actual decay rate is determined by the emotion type and the agent's personality such that with agreeable agents, negative emotions decay faster than positive ones.

2.2 Emotion Regulation

In their seminal work on non-verbal behavior, Ekman and Friesen [6] argue that the expression of emotional states (e.g., as facial expression) is governed by social and cultural norms, so-called *display rules*, that have a significant impact on the intensity of emotion expression. We will treat emotion regulation as a process that decides whether an emotion is expressed or suppressed. Moreover, a value is calculated that indicates to what extent an emotion is suppressed. An agent's emotion regulation is depending on a multitude of parameters [14, 5]. We broadly categorize them into parameters that constitute a social threat for

the agent, and parameters that refer to the agent's capability of (self-)control. Although this distinction is somewhat arbitrary, we found that it allows authors to state regulation parameters in a simple and intuitive way.

Communication is always embedded into a social context where participants take social roles with associated communicative conventions. Following Brown and Levinson [3], we take *social power* θ_P and *social distance* θ_D as the most important social variables ($\theta_P, \theta_D \in \{0, \ldots, 5\}$). We assume that roles are ordered according to a power scale, where *social_power(L2,L1,*θ_P*,Sit)* means that agent $L2$ is θ_P ranks higher than agent $L1$. Social distance refers to the familiarity or 'closeness' between agents, and can be stated as *social_distance(L1,L2,*θ_D*,Sit)*. Based on θ_P and θ_D, the *social threat* θ for $L1$ from $L2$ is computed as $\theta = \log_2\left(2^{\theta_P} + 2^{\theta_D}\right)$. If θ_P and θ_D are both zero, θ is set to zero. Note that the social variables are not meant to reflect 'objective' ratings of power or distance, but the modelled agent's assumed assessment of the ratings.

The following set of parameters describe the agent's *self-control* each of which takes a value $\gamma_i \in \{-5, \ldots, 5\}$. Greater positive values indicate that the agent is capable and willing to suppress negative emotions whereas greater negative values indicate that the agent tends to also express negative emotions. Besides the agent's agreeableness, we also consider the *extroversion dimension of personality*. Extrovert agents typically express their emotions independent of their impact on another agent whereas introvert agents tend to refrain from doing so. For artistic reasons, we discourage authors from using the zero value, since agents with 'neutral' personality might fail to express their emotions succinctly. Moreover, if the agent assumes that the *interlocutor's personality* is unfriendly (disagreeable), it will rather not express a negative emotion. An interesting phenomenon in interactions among humans are *reciprocal feedback loops* where one agent's linguistic friendliness results in the interlocutor agent's adaption of its otherwise unfriendly behavior.

The overall control value γ is computed as $\gamma = \frac{\sum_i \gamma_i}{N}$ where the denominator N scales the result according to the number of considered control parameters. Basically, the equation captures the intuition that different control parameters may defeat each other. Thus, the control of an agent that is very extrovert but deals with a very unfriendly interlocutor might be neutralized to some degree.

The **(Social) Filter Module** operates on the winning emotional state, the social threat, and the overall control value. It outputs an *external emotion* with a certain intensity $\epsilon \in \{0, \ldots, 5\}$, i.e., the *type* of emotion that will be displayed by the agent. The Filter module consists of only two rules, one for positive and one for negative emotions. The general form of a social filter rule is as follows.

external_emotion(L1,L2,E,ϵ,Sit) **if** social_threat(L1,L2,θ,Sit) **and**
control(L1,L2,γ,Sit) **and**
winning_emotional_state(L1,L2,E,δ,Sit)

The most difficult problem here is to adequately combine the intensity values associated with the social threat experienced by the agent, the agent's control capability, and the emotional state. The default combination function for neg-

ative emotions is $\epsilon = \delta - (\theta + \gamma)$. Intuitively, the function balances the social threat against the agent's control, whereby high values for threat may neutralize the lacking self-control of the agent to a certain extent. The filter rule for positive emotions is syntactically identical but uses a different combination function: $\epsilon = \delta - (\theta - \gamma)$. Here, it is the agent's low control that dominates the expression of emotions. Alternatively, we provide a decision network to determine whether and to what extent an agent expresses its emotional state, based on its check for negative consequences of emotion expression.

2.3 Emotion Expression

External emotions must eventually be described in terms of the agent's reactions and behaviors. We use a simplified version of Ortony's categorization of emotion response tendencies [12], and distinguish between expressive and information-processing responses. *Expressive responses* include somatic responses (flushing), behavioral responses (fist-clenching, throwing objects), and two types of communicative responses, verbal and non-verbal (e.g., frowning). *Information-processing responses* concern the agent's diversion of attention and evaluations (which we handle in the Affect Processing module). The Animation Engine currently used only allows for rather crude forms of combining verbal and non-verbal behavior [10]. Body movements (including gestures) may precede, overlap, or occur subsequently to verbal utterances. An interesting alternative is the BEAT system [4] that autonomously suggests appropriate gestures for given speech.

2.4 Affect Processing

The Agent Model describes an agent's mental state. We distinguish *static* and *dynamic* features of an agent's mind state, such that the agent's personality and standards are considered as static whereas goals, beliefs, attitudes and social variables are considered as dynamic. Here, we are mainly concerned with change of attitude as a result of social interaction.

Ortony [11] suggests the notion of *(signed) summary record* to capture our attitude toward or dispositional (dis)liking of another person. This record stores the sign of emotions (i.e., positive or negative) that were induced in the agent L by an interlocutor I together with emotions' associated intensities. In order to compute the current intensity of an agent's (dis)liking, we simply compare the (scaled) sum of intensities of elicited positive and negative emotions (δ^σ, $\sigma \in \{+, -\}$), starting in situation $Sit_0^{L,I}$, the situation when the interaction starts. We will only consider the intensity of the winning emotional state δ_w. If no emotion of one sign is elicited in a situation, it is set to zero.

$$\delta^\sigma(Sit_n^{L,I}) = \frac{\sum_{i=0}^{n} \delta_w^\sigma(Sit_i^{L,I})}{n+1}$$

Positive values for the difference $\delta^+ - \delta^-$ indicate an agent's liking of an interlocutor and negative ones indicate disliking. The more interesting case where an

interlocutor the agent likes as a consequence of consistent reinforcement (suddenly) induces a high-intensity emotion of the opposite sign, e.g., by making the agent very angry, is captured by the following update rule.

$$\delta(Sit_n^{L,I}) = \delta^{\sigma}(Sit_{n-1}^{L,I}) \times \omega_h \mp \delta_w^{\overline{\sigma}}(Sit_n^{L,I}) \times \omega_r$$

The weights ω_h and ω_r denote the weights we apply to historical and recent information, respectively. ω_h and ω_r take values from the interval $[0, 1]$ and $\omega_h + \omega_r = 1$. A greater weight of recent information is reflected by using a greater value for ω_r. As to the question how the obtained (dis)liking value affects future interactions with the interlocutor, two interpretations are considered. While *momentary (dis)liking* means that the new value is active for the current situation and then enters the summary record, *essential (dis)liking* results in the new value replacing the summary record.

3 MPML: A Markup Language for Character Control

We currently use the Microsoft Agent package [10] as our Animation Engine, which allows to embed animated characters into a web page based JavaScript interface. The package comes ready with controls for animating 2D cartoon-style characters, speech recognition and a Text-to-Speech (TTS) engine. In order to facilitate the process of scripting more complex scenarios, including, e.g., sequential and parallel activity of multiple characters, we have developed an XML-style markup language called MPML (Ishizuka *et al.* [8]).

Basic tagging schemes for a character's behavior and multi-character coordination are `<act/>` where a character performs a pre-defined animation sequence ("alert", "blink", "decline", "explain", "greet", "sad", "suggest", etc.); `<speak>...</speak>` where a character speaks a pre-defined sentence which is also displayed in a balloon; `<listen>...</listen>` where the character is prepared to recognize pre-defined user utterances; `<seq>...</seq>` for sequential behavior of multiple characters, and `<par>...</par>` for parallel behavior of multiple characters.

In short, MPML is a powerful and easy-to-use markup language that allows content authors to script rich web-based scenarios featuring animated characters. Typically, MPML is used to design characters with scripted behaviors, i.e., the author has full control over a character's verbal and non-verbal behavior. However, the restriction to scripted behavior can be relaxed by interfacing MPML with SCREAM's reasoning module that supports autonomous control of a character's affective behavior. Communication between MPML and the Java applet (driving SCREAM by Java-to-Jinni and Jinni-to-Java method calls) is realized by special tagging schemes. The `<execute/>` tag may call a Java method, e.g., to assert a communicative act of another agent to the character's knowledge base. The `<consult>...</consult>` tagging scheme together with the child tagging scheme `<test>...</test>` is used to retrieve the character's reaction from SCREAM. Depending on the value of the `test` element, the character will perform a sequence of verbal and non-verbal behaviors.

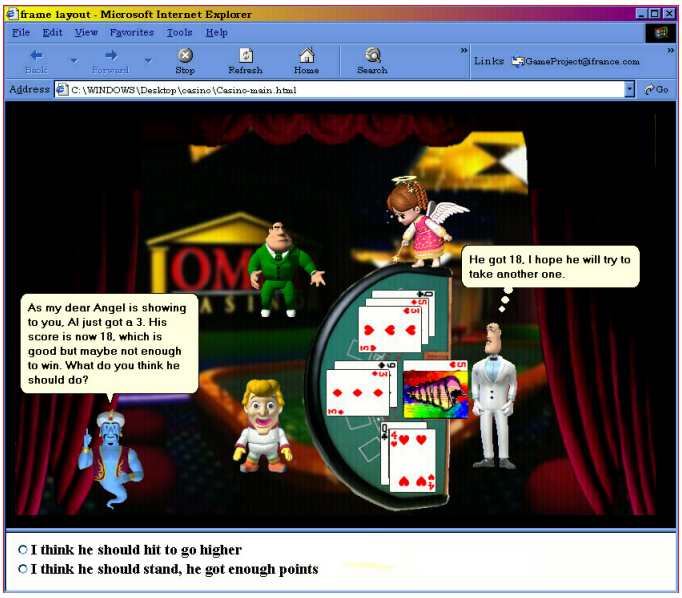

Fig. 2. Casino Scenario.

4 Illustrative Example

In this section we will illustrate how our system works. As an interaction setting, we choose a casino scenario where a user and other characters can play the "Black Jack" game. Fig. 2 shows the situation where the character "Genie" practices Black Jack with the user by commenting the game of character "Al" (Genie is the character at the bottom-left of the Internet Explorer window, and Al is the male character to the right of the dealer).

We will now watch the user playing five games of Black Jack and thereby demonstrate how Genie's mental make-up as well as the (affective) interaction history determine his behavior. For expository reasons, we let the user *never* follow Genie's advice, and we use a very sparse Agent Model. Among others, Genie is assumed as rather agreeable and extrovert, he is socially close to the user and also (initially) slightly likes the user. His goals are that the user wins (with low intensity), and that the user follows his advice (with high intensity). Note that the outcome of the the game, i.e., whether the user wins or looses, is independent of her or him following Genie's advice.

- In the **first game** (user looses) Genie's winning emotional state is *distress* with intensity 4, because the user did not follow his advice. However, he displays *distress* with low intensity as his agreeable personality effects a decrease in the intensity of negative emotion expression.
- In the **second game** (user looses) Genie is sorry for the user with intensity 4, since positive ('sorry for' the user's lost game) emotions decay slowly

and sum up, which leads to an increase in Genie's liking of the user. His personality traits let him express the emotion with even higher intensity.
- In the **third game** (user looses) Genie gloats over the user's lost game, because at that point, the negative emotions dominate the positive ones as a consequence of the user's repeated refusal to follow Genie's advice. Hence Genie's attitude changes to slightly disliking the user which lets him experience *joy* over the user's *distress* (*gloat* with intensity 5). Again, Genie's friendly personality decreases the intensity of the external emotion.
- In the **fourth game** (user wins) Genie's emotional state is *bad mood* with intensity 5, slightly more than his *happy for* emotion (as the user wins the game this time). Here an overall, unspecific affective state (mood) is expressed with low intensity, rather than a specific emotion.
- In the **fifth game** (user wins) Genie's dominant emotional state is *resent* with intensity 4, because he slightly dislikes the user and consequently is distressed that the user won although she or he ignored his advice. Genie expresses his emotion with reduced intensity.

An exhaustive exploration of all possible interaction patterns in the described game scenario reveals that Genie's reactions are conform at the beginning games and show more variety in the subsequent games. This can be explained by the evolution of Genie's attitude toward the user, depending on whether the user follows or refuses to follow Genie's advice. In effect, Genie's attitude decides, e.g., whether he is *sorry for* or *resents* the user's lost game. However, in accordance with Genie's agreeableness, his emotional reactions are mostly positive.

5 Conclusion

Recent years have witnessed a growing interest in life-like, believable characters, as they might be a crucial component of enhanced learning and presentation systems. Although it is widely recognized that emotion and personality are key factors for characters' believability, tools that facilitate the autonomous generation of affective behavior are still rare. Notable exceptions are [15, 1, 14, 5].

In this paper, we discuss models and tools for scripting and coordinating affective interactions with and among animated believable characters. While MPML is a powerful tool for controlling and coordinating the visual behavior of characters (their 'body'), the SCREAM system constitutes a practical technology for scripting the mental processes underlying a character's affective behavior (its 'mind'). Most importantly, it is more flexible than systems with a similar aim [15, 1, 5] as the author may decide on the level of detail at which the character is scripted (the 'granularity' feature). If many levels of indirection of the agent's behavior are desirable, the author may define all of the available parameters and also control the influence of each parameter by editing the combination functions. In certain settings, however, only a subset of the parameters might be of interest, e.g., when the author wants to script a (interactive) presentation agent that is only driven by goals and personality. The system will manage the elicited emotions and produce an output that reflects the provided influences.

Acknowledgments

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

References

1. E. André, T. Rist, S. van Mulken, M. Klesen, and S. Baldes. The automated design of believable dialogue for animated presentation teams. In J. Cassell, S. Prevost, J. Sullivan, and E. Churchill, editors, *Embodied Conversational Agents*, pages 220–255. The MIT Press, 2000.
2. BinNet Corp. *Jinni 2000: A high performance Java based Prolog for agent scripting, client-server and internet programming*, 2000. URL: `www.binnetcorp.com`.
3. P. Brown and S. C. Levinson. *Politeness. Some Universals in Language Usage*. Cambridge University Press, 1987.
4. J. Cassell, H. Vilhjálmsson, and T. Bickmore. BEAT: the Behavior Expression Animation Toolkit. In *Proceedings of SIGGRAPH-01*, pages 477–486, 2001.
5. B. de Carolis, C. Pelachaud, I. Poggi, and F. de Rosis. Behavior planning for a reflexive agent. In *Proceedings 17th International Conference on Artificial Intelligence (IJCAI-01)*, 2001.
6. P. Ekman and W. V. Friesen. The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *Semiotica*, 1:49–98, 1969.
7. B. Hayes-Roth and P. Doyle. Animate characters. *Autonomous Agents and Multi-Agent Systems*, 1(2):195–230, 1998.
8. M. Ishizuka, T. Tsutsui, S. Saeyor, H. Dohi, Y. Zong, and H. Prendinger. MPML: A multimodal presentation markup language with character control functions. In *Proceedings Agents'2000 Workshop on Achieving Human-like Behavior in Interactive Animated Agents*, pages 50–54, 2000.
9. W. L. Johnson, J. W. Rickel, and J. C. Lester. Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, 11:47–78, 2000.
10. Microsoft. *Developing for Microsoft Agent*. Microsoft Press, 1998.
11. A. Ortony. Value and emotion. In W. Kessen, A. Ortony, and F. Craik, editors, *Memories, thoughts, and emotions: Essays in the honor of George Mandler*, pages 337–353. Hillsdale, NJ: Erlbaum, 1991.
12. A. Ortony. On making believable emotional agents believable. In R. Trappl, P. Petta, and S. Payr, editors, *Emotions in Humans and Artifacts*. The MIT Press, 2001.
13. A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.
14. H. Prendinger and M. Ishizuka. Social role awareness in animated agents. In *Proceedings 5th International Conference on Autonomous Agents (Agents-01)*, pages 270–277, 2001.
15. W. S. N. Reilly. *Believable Social and Emotional Agents*. PhD thesis, Carnegie Mellon University, 1996. CMU-CS-96-138.
16. D. Rousseau and B. Hayes-Roth. A social-psychological model for synthetic actors. In *Proceedings 2nd International Conference on Autonomous Agents (Agents-98)*, pages 165–172, 1998.
17. S. J. Russell and P. Norvig. *Artificial Intelligence. A Modern Approach*. Prentice Hall, Inc., Upper Saddle River, New Jersey, 1995.