Künstliche Intelligenz (KI) Zeitschrift, Vol.1, 2004, pages 4-10

Scripting and Evaluating Affective Interactions with Embodied Conversational Agents

Helmut Prendinger, Junichiro Mori, Santi Saeyor, Kyoshi Mori, Naoaki Okazaki, Yustinus Juli, Sonja Mayer, Hiroshi Dohi, and 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: prendinger@acm.org

Abstract

This paper describes the results obtained and ongoing agenda of a research project on embodied conversational agents, carried out at the University of Tokyo. The main focus points of the project are the development of scripting languages for controlling life-like agents and the modeling of affective interactions between agents and human users. Furthermore, the project aims at evaluating the impact of character-based interfaces on the emotional state of users. In this paper, we will explain and illustrate a selection of major project results.

1 Introduction and Motivation

Reflecting the importance and promise of animated interface agents for human-computer interaction, the Japanese Society for the Promotion of Science (JSPS) launched the project "Multimodal Anthropomorphic Interface and the Foundations of its Intuitive and Affective Functions" as part of the Future Program ("Mirai Kaitaku") in 1999, as a five-year project under the lead of Mitsuru Ishizuka (University of Tokyo). The purpose of this paper is to present the major goals and results of the project as well as ongoing and future research issues related to the project.

The objectives of the project were two-fold: first, to develop markup languages that allow for easy control of the behavior of synthetic embodied agents in a web browser; and second, to increase the believability of synthetic agents by providing them with affective functions such as emotion and personality. The first objective was achieved by designing XML-compliant languages that offer easy-to-use tagging structures to coordinate the verbal and nonverbal behavior of multiple embodied agents, and integrate them into the web environment. As a result, the Multi-modal Presentation Markup Language (MPML) has been developed. Rather than being a single language, MPML refers to a family of markup languages, where each member has its particular focus and strength, but all members share the aim to address non-technically oriented web content designers who want to include embodied agents into their web site. Accordingly, one version of MPML also provides a visual editor to facilitate the generation of a presentation script.

The second objective was met by basing character behavior on models of emotion and personality, and findings from socio-psychological studies. This resulted in the development of the Scripting Emotion-based Agent Minds (SCREAM) system,

a mechanism that allows content authors to design agents that autonomously generate emotionally and socially appropriate behaviors depending on their mental make-up. In SCREAM, a character's mental state is determined by its goals, beliefs, attitudes, affect-related features of the interlocutor's behavior, and parameters peculiar to the social interaction context. A high-level declarative language (Prolog) is used encode the character's profile, and interfaced with (one version of) MPML.

Although the integration of MPML and SCREAM facilitated the authoring of interactive presentations employing embodied agents that display affective behavior, it remained unclear how users perceive character-based interfaces. Therefore a simple experiment has been performed that uses bio-signals to determine the impact of agents with affective display on the emotional state of users.

The remainder of the paper is organized as follows. In Section 2, we will first describe the MPML family and a visual editor for one version of MPML, and then illustrate the markup language by means of an example. Section 3 provides a condensed explanation of the SCREAM system. In Section 4, we report on an experiment with a character-based quiz game. In Section 5, we briefly describe ongoing and future work. Section 6 summarizes and concludes the paper.

2 The MPML Family of Character Control Languages

MPML (Multi-modal Presentation Markup Language) is a language specifically designed for non-expert content authors that enables to direct the behavior of multiple embodied characters in a web environment. First, MPML is a *markup language* compliant with standard XML and hence allows for scripting in a style that is familiar to a broad audience (assuming some background with HTML scripting). Second, MPML is a language designed with the aim of scripting character-based *presentations* that can be viewed in a web browser. In order to facilitate the generation of different types of presentations, including interactive presentations, MPML provides tagging structures that enable authors to utilize features of presentations given by human presenters in web-based presentations environments, such as dynamic media objects or interaction with the audience. Finally, MPML supports the generation of *multi-modal* presentations, that is, presentations utilizing multiple mechanisms to encode the information to be conveyed, including 2D and 3D graphics and spoken (synthetic) language, music, and video (Bordegoni et al. 1997). Our particular focus has been the modalities specific to embodied agents. Besides synthetic speech, the agents may communicate information by using multiple modalities, such as facial displays in order to express emotions, hand gestures including pointing and propositional gestures, head movements ("nodding"), and body posture.

While animating the visual appearance of embodied characters is a difficult task that involves many levels – from changes to each individual degree of freedom in the motion model to high-level concerns about how to express a character's personality by means of its movements – we largely sidestepped those problems by using the Microsoft Agent package as our animation engine (Microsoft 1998). This package provides controls to animate 2D cartoon-style characters, a text-to-speech engine and voice recognizer. Characters controlled by the Microsoft Agent package may perform pre-defined animation sequences, including animations for "alert", "decline", "explain", "greet", "sad", and so on. Each character has approximately fifty animations available.

Most other existing scripting languages cover a range of different "abstraction levels" in a single language. The Character Markup Language (CML) developed by Arafa et al. (2002) allows specifying high-level concepts such as the emotion "happy" as well as low-level behaviors like "blinking". The Virtual Human Markup Language (VHML) developed by Mariott and Stallo (2002) comprises tags for facial and body animation, speech, gesture, and even dialogues. Scripting Languages also differ in their focus on a particular competence envisioned for the character. The Behavior Expression Animation Toolkit (BEAT) of Cassell et al. (2001) provides sophisticated synchronization of synthetic speech and nonverbal behavior, and the Affective Presentation Markup Language (APML) of De Carolis et al. (2002) targets communicative functions.

2.1 Two Types of MPML Character Control Languages

In the project, two types of markup languages for character and presentation control have been developed. In the *converter-type* MPML languages, an application program transforms the MPML script file to a script that is executable in a web browser (JavaScript). In the *XSL-based* MPML languages, the "eXtensible Stylesheet Language" (XSL) is employed to define the form of the MPML content script. Figure 1 shows a screenshot where an embodied agent presents some members of the MPML family. The following three languages adhere to the technique of using an XSL stylesheet to convert the MPML script file to JavaScript 'on the fly'.

- *MPML2.2a*: This version supports sequential and parallel behavior of multiple embodied agents. It provides an interface to Macromedia Flash, so that agents can control a Flash movie and a Flash movie may trigger agent behavior (Saeyor 2002).
- *DWML*: This member of XSL-based MPML languages is concerned with scripting time-dependent relations between web-based media objects in addition to displaying an animated agent. The so-called Dynamic Web Markup Language (DWML) supports the following media objects: dynamic text, graphics, audio, and video (Du and Ishizuka 2001). Here, a time control function enables to define the temporal sequence of media objects during presentation, which are otherwise (in HTML/JavaScript programming) immediately shown when a web page is loaded.
- *MPML-VR*: This language version is a variant of MPML2.2a tailored to control a 3D virtual space and a 3D agent. The resulting markup language – MPML for Virtual Reality – allows for presentations in three-dimensional space (Okazaki et al. 2002).

The language we will discuss in some detail in the next section, MPML3.0, is a descendent of earlier developed converter-type markup languages that employ the Microsoft Agent package. This version also allows directing rich affective expression of a pseudo-muscle based 3D "talking head" (see Fig. 2), which is described in Barakonyi and Ishizuka (2001).

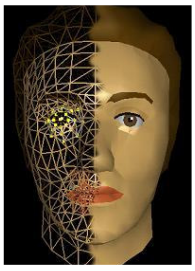

Figure 2 The "SmArt" agent.

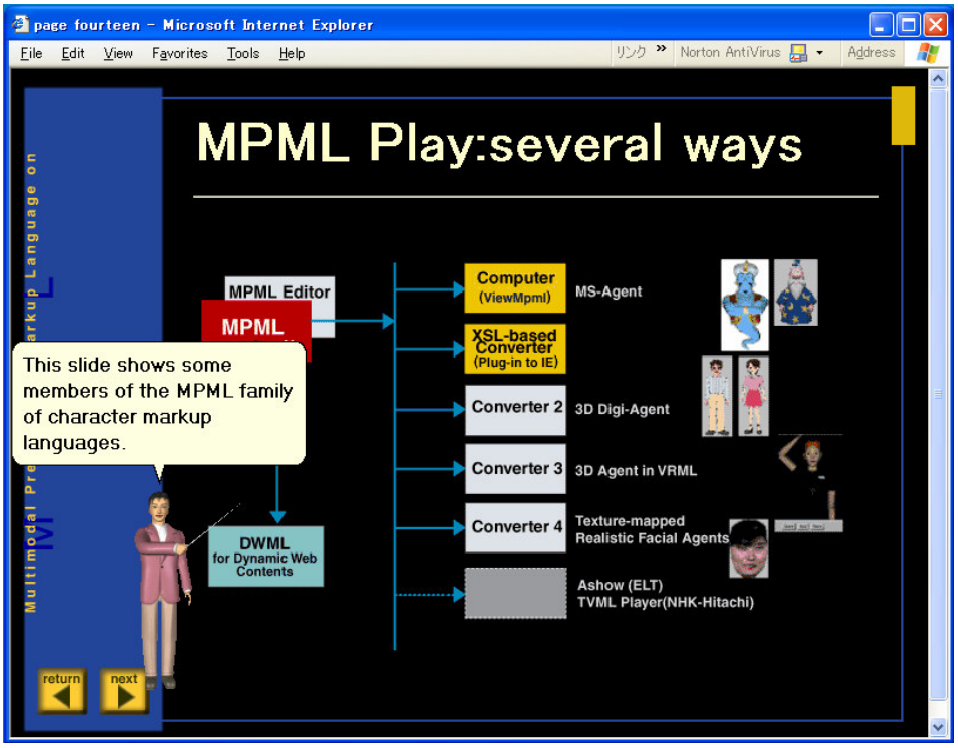

Figure 1 The „Shima“ character presents control technologies for embodied agents.

2.2 MPML3.0 Visual Editor

Script authors who work with MPML3.0 may edit the file containing the tagging structures for character and presentation control. However, considering the complexity of a presentation script with deeply nested tagging structures and the popularity of visual interfaces, manipulating a graphical representation (corresponding to the script) is often preferable. For this reason, the MPML3.0 Visual Editor has been developed. The Visual Editor is an application program which integrates four modules.

- The *Script Loader* module loads the text file containing the MPML script and checks the script for syntactical errors.
- The *Graph* module visualizes the script by generating a graphical presentation of the presentation.
- The *Script Saver* module converts the graph to a textual MPML script.
- The *Converter* module transforms the MPML script to JavaScript.

The resulting“control web page” instructs the characters’ behavior and background web pages, that is, pages depicting the environment the characters inhabit.

The Visual Editor consists of two main windows (see Fig. 3). The window to the left – the (presentation) *Graph* window – shows the graphical presentation of the MPML script, and the window to the right – the *Current Mode* window – displays the current location of user interaction with the graph. The upper part of the Current Mode window allows the script author to choose a character (for instance, “Marge”), the intended character behavior, such as “act” (perform an animation), “speak” (an

utterance), or “move” (to a certain location on the screen), and the web page that serves as a background for the agents’ performance. The lower part of the Current Mode window depicts the current attribute-value pair of the element whose associated box (configuration) in the Graph window shares the physical location with the current mouse position. Authors may edit tag elements in the Current Mode window and then “drag and drop” the box associated with the tag at the appropriate position in the presentation graph in the Graph window.

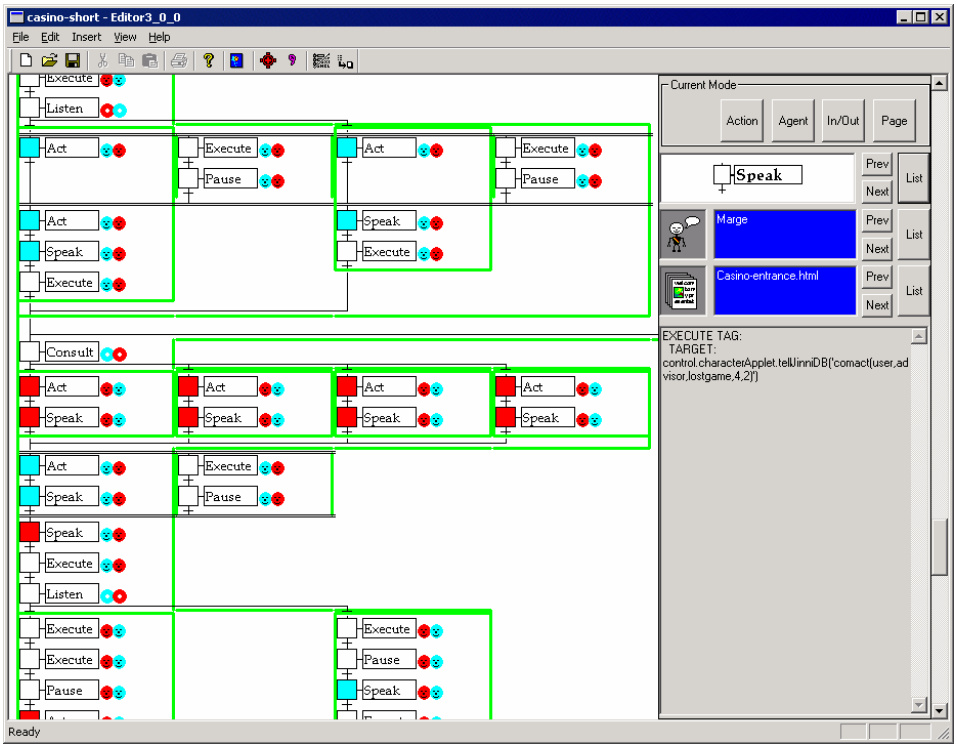

Figure 3 MPML3.0 Visual Editor.

A presentation graph is built up from the following entities. A *node* is displayed as a box (configuration) that essentially refers to a tag element. The *edges* in the graph can be divided into three types. A *sequential edge* is a (down-side) directed arc between two nodes and denotes the next event in the presentation flow. A *parallel edge* is a directed arc between a node and a set of nodes each of which initializes a sequence of events, in particular actions carried out by multiple agents in parallel. A *branching edge* is a directed arc between a node and a set of nodes such that the node that satisfies a certain condition initializes a sequence of events. The condition may depend on user interaction or, if autonomous agents are used (see Section 3), the behavior suggested by the reasoning engine of the agent.

2.3 Illustration

In this section, we show how an author may use MPML to mark up an interactive web-based presentation. As an interaction setting, we will describe a casino scenario where the user and another embodied agent (“Al”) play the “Black Jack” game

against the dealer “James” (see Fig. 4). At the bottom left part of the window, the character “Genie” (called “Djinn”) acts as the user’s advisor to play the game.

The casino scenario employs two types of character control paradigms. While the behaviors of James and Al are pre-defined (“scripted”), a knowledge base encoding affective reasoning processes autonomously generates the responses of the advisor Djinn. Hence, characters that play a minor role in the scenario can be scripted in a straightforward way and the scenario author may focus on the specification of characters whose emotional reaction is relevant to the development of the interaction. Observe that the presentation graph describing the space of possible traversals (instances of presentations) is fixed. André et al. (2000) follow a different approach where a dialogue between multiple characters is automatically generated. Here a central planner assigns dialogue contributions to presentation agents depending on their role in the scenario and models of emotion and personality.

Figure 4 Casino Scenario.

By means of the following script (sketch) we will explain some of the tagging structures of MPML3.0.

```
1  <listen agent=”Genie”>
2    <heard value=”Djinn I hit”>
3    <scene agents=”Genie,James”>
4      <page ref=”casino-main.html”>
5        <execute target=”control.changebottom(‘ans.html’)”/>
6        <execute target=[…].href=’g3-2.html’”/>
```

```
7     <act agent="Genie" act="Uncertain"/>
8     <speak agent="Genie">
9       We have 17 now…That's not an easy decision, but I would stand.
10    </speak>
11    <execute target="control.changebottom('ans-5.html')"/>
12    <listen agent=Genie">
13      <heard value="Stand">
14        <scene agents="James,Genie">
15          <page ref="Casino-main.html">
16            <par>
17              <seq><act agent="James" act="GestureRight"/></seq>
18              <seq><execute target="[…].href='g3-3win.html'"/></seq>
19            </par>
20            <act agent="James" act="Sad"/>
21            <speak agent="James">
22              <emotion assign="sadness"/>
23              The bank gets 24 and looses. Player wins, you're lucky guys.
24            </speak>
25            <execute target="control.chApplet.tellJinniDB(
26                        'comact(user,advisor,wongame,4,1)')"/>
27          </page>
28        </scene>
29      </heard>
30      <heard value="Hit">
31        …
32      </heard>
33      …
34  </heard>
35  …
36 </listen>
```

In line 1, the character "Genie" is enabled to accept the speech command from the user, followed by a branching edge of multiple alternatives, where the first branch is partly shown in lines 2-34. This branch is chosen when the user utters: "Djinn I hit". First, in line 5, a frame window denoted by ans.html replaces the bottom frame in order to temporarily disable user interaction. Next, in line 6, a (embedded) sub-frame window of the main window (casino-main.html) is replaced by the new sub-frame window g3-2.html that depicts the updated game board state. After Djinn is starting to display the "Uncertain" animation (line 7), he suggests to "stand" (lines 8-10), and then the bottom frame is being replaced by a sub-frame window (ans-5.html) depicting a new pair of choices (line 11).

Let us now describe the expansion of the branch where the user decides to follow Djinn's suggestion (lines 13-29). Lines 16-19 encode the parallel execution of two actions. The dealer performs the "GestureRight" animation (line 17) in order to demonstrate the new game situation, which is simultaneously loaded (line 18). Then the dealer nonverbally (line 20) and verbally (line 21-24) expresses his sadness that he lost this round of the game. Here we use the emotion element (an empty tag) to modulate speech output, following the description of the vocal effects associated with five emotions investigated by Murray and Arnott (1995). In line 25-26, the execute tag is used to update Djinn's knowledge base, telling that the user won the current round, which is internally represented by (round) 4, (choice) 1. At this point, MPML

interacts with the SCREAM system that is used to derive Djinn's affective response. SCREAM will be briefly discussed below. The remaining lines 27-36 show some of the required closing tags.

3 Designing Emotion-based Agents

A task complementary to scripting the visual appearance of a character is to author the character's mental state and emotional reaction to its environment. We have developed a system called SCREAM (SCRipting Emotion-based Agent Minds) that facilitates scripting a character's affect-related processing capabilities (Prendinger et al. 2002). The system allows to specify a character's mental make-up and to endow it with emotion and personality that are considered as key features for the life-likeness of characters. A character's mental state can be scripted at many levels of detail (granularity levels), 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. For portability and extensibility, the SCREAM system is written in Java and Jinni, a Java based Prolog system (BinNet Corp. 2003).

The SCREAM system is in many respects similar to Reilly's (1996) Em architecture, but more flexible in the sense of allowing for granular scripting. Like the Extempo (2003) and IMP (André et al. 2000) systems, SCREAM exploits web technologies so that emotion-based embodied agents can be run in a web browser. Although the system supports authoring character ensembles, it does not do so automatically, as done by André et al. (2000). The main reasons are that we wanted to give the author full control over each dialogue move and delegate the task of producing the propositional content – as opposed to "affective rendering" – of the agents' communicative acts to the application designer.

The following paragraphs provide a quick walk through the main components of the SCREAM system: Emotion generation, emotion regulation and expression, and the agent model (for extensive discussion, see Prendinger et al. 2002). A core activity of an emotion-based agent mind is *emotion generation* and the management of emotions, which is dealt with by three modules, the appraisal module, the emotion resolution module, and the emotion maintenance module. Reasoning about emotion models an agent's appraisal process, where events are evaluated as to their emotional significance for the agent (Ortony et al. 1988). The significance is determined by so-called "emotion-eliciting conditions", the agent's beliefs, goals, standards, and attitudes. Emotion types can then be seen as classes of eliciting conditions, each of which is labeled with an emotion word of phrase, such as joy, distress, "happy for", "sorry for", and so on. All emotions have associated intensities depending on the intensities of its conditions. Since a reasonably interesting agent will have a multitude of mental states (beliefs, goals, attitudes, etc.), more than one emotion is typically triggered when the agent interacts with another agent. The emotion resolution and maintenance modules determine the most dominant (winning) emotion and handle the decay process of emotions, respectively.

The expression of emotions is governed by social and cultural norms that have significant impact on the intensity of their expression. We will treat *emotion regulation* as a process that decides whether an emotion is expressed or suppressed (Prendinger and Ishizuka 2001). We categorize regulatory (control) parameters into ones that constitute a social threat for the agent (social distance and social power), and parameters that refer to the agent's capability of (self-)control (personality,

interlocutor personality, and linguistic style). An overall control value, computed from the given (possibly mutually defeating) control values, determines the intensity of expression of the elicited emotion.

The *agent model* describes the mental state of an agent. We distinguish static and dynamic features of an agent's mind state, such that the agent's personality (agreeableness, extroversion) and standards are considered as static whereas goals, beliefs, attitudes and social variables are considered as dynamic. One main concern has been the change of an agent's attitude as a result of social interaction, based on Ortony's (1991) *(signed) summary record* of dispositional (dis)liking. This record stores the sign (positive or negative) and intensity of emotions that were induced in the agent by an interlocutor. In effect, attitudes not only contribute to the elicitation of emotions by deciding whether the agent has a "sorry for" or "gloat" emotion – but induced emotions may also change an agent's affective state, in particular, its attitude and familiarity toward another agent.

4 Evaluating the Effects of Embodied Conversational Agents

This part of the project aims to show the impact of embodied conversational agents on the emotional state of human users (Mori 2003). Interacting with computers is often responsible for negative emotional states of the user, such as frustration or stress. One way to alleviate the intensity of user frustration is to provide appropriate feedback. As people tend to respond to computers in an essentially natural way (Reeves and Nass 1998), we suggest using an interface agent that gives affective feedback including the expression of empathy. In order to measure the effect of the agent's response on user emotions, we take physiological signals from the user.

Our work follows up to recent studies in the *Affective Computing* paradigm (Picard 1997) that suggest employing bio-signals to detect user emotions (Schreier et al. 2002) and affective feedback to reduce (deliberately induced) user frustration (Klein et al. 2002). By contrast, we employ an embodied interface agent rather than a text-based interface to communicate with the user. This design choice may also shed new light on the *Persona effect*, which relates (among others) to the credibility and motivation enhancing effects of character-based interfaces (Lester et al. 1997). Unlike standard evaluation methods such as questionnaires, the use of physiological data may allow for a more precise assessment of users' perception of the interface.

4.1 Design of the Experiment

We implemented a simple mathematical quiz game where subjects are instructed to sum up five successively displayed numbers and are then asked to subtract the i-th number of the sequence ($i \leq 4$). Subjects compete for the best score in terms of correct answers and time (a monetary award was given for both participation and best score). Subjects were told that they would interact with a prototype interface that might still contain some bugs. Before game start, the "Shima" character shows some quiz examples that explain the game. This period also serves to collect physiological data of subjects that are needed to normalize data obtained during game play. In six out of a total of thirty quiz questions, a delay was inserted before showing the 5^{th} number. The delay, about 9 seconds on average, is assumed to induce frustration as the subjects' goals of giving the correct answer and achieving a fast score are thwarted, called "primary frustration" in behavioral psychology (Lawson 1965).

In the experiment, subjects were twenty male students, all of them native speakers of Japanese. We randomly assigned subjects to one of two versions of the game (ten in each version).

- *Affective version*. Depending on whether the subject selects the correct or wrong answer from the menu displayed in the game window (see left part of Fig. 5), the agent expresses "happy for" and "sorry for" emotions both verbally and nonverbally. If a delay in the game play happens, the agent expresses empathy for the user after the subject answers the question that was affected by the delay.
- *Non-affective version*. The agent does not give any affective feedback to the subjects. It simply replies "right" or "wrong" to the user's answer and does not comment on the occurrence of the delay.

Figure 5 shows the agent expressing empathy to the user since a delay occurred. The agent displays a gesture that Japanese people perceive as a signal of the interlocutor's apology, and says: "I apologize that there was a delay in posing the question" (English translation). Note that the apology is given *after* the occurrence of the delay, immediately after the subject answers the question.

Subjects are attached to two types of sensors, skin conductivity (SC) and blood volume pulse (BVP) on the first three fingers of their non-dominant hand. Since SC co-varies with the level of arousal, and heart rate (automatically calculated from BVP) with negative valence of emotion (Picard 1997), the signals can be used to infer user emotions as a location in the valence-arousal space of emotion (Lang 1995). Signals are recorded via the ProComp+ unit and visualized using Thought Technology software.

In order to show the effect of the agent's behavior, we have been interested in three specific segments. The DELAY segment refers to the period after which the agent suddenly stops activity while the question is not completed until the moment when the agent continues with the question. The DELAY-RESPONSE segment refers to the period when the agent expresses empathy concerning the delay, or ignores the occurrence of the delay – which follows the agent's response (regarding the correctness of the answer) to the subject's answer. The RESPONSE segment refers to the agent's response to the subject's correct or wrong answer to the quiz question.

4.2 Results of the Experiment

The first observation relates to the use of delays in order to induce frustration in subjects. All eighteen subjects showed a significant rise of SC in the DELAY segment, indicating an increased level of arousal. The data of two subjects of the non-affective version were discarded because of extremely deviant values. Since the BVP data of only six out of twenty subjects could be taken reliably, our hypotheses below are only based on SC data.

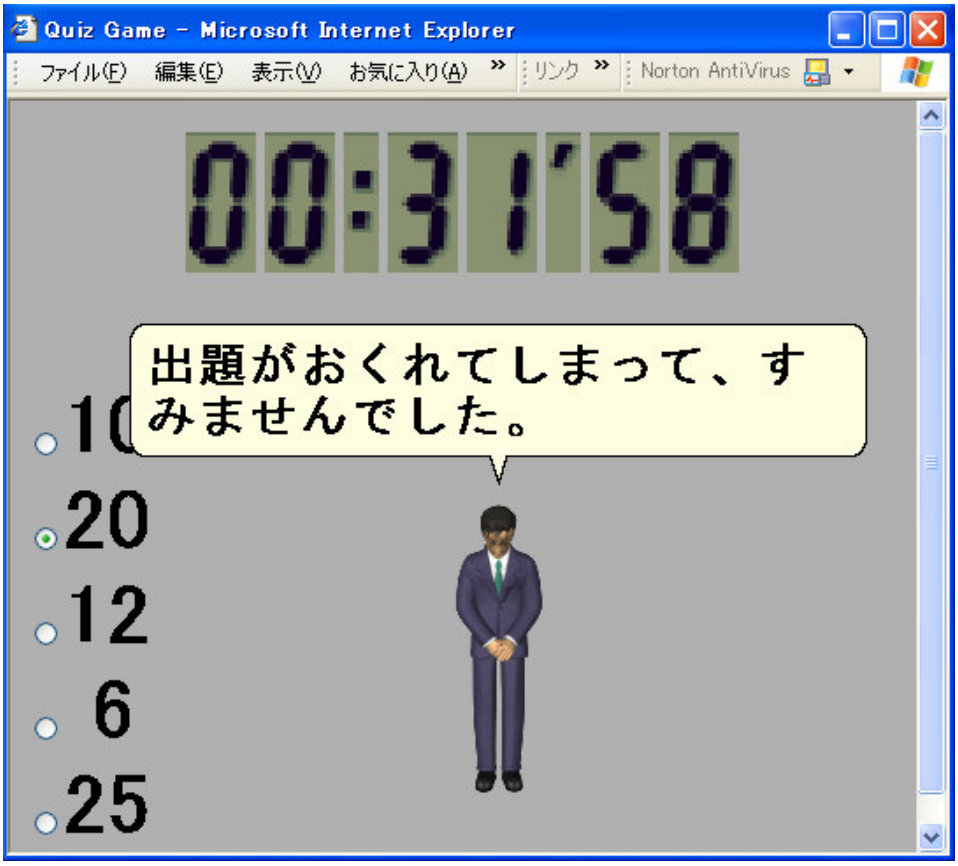

Figure 5 The "Shima" agent apologizes for the delay, by saying "I apologize that there was a delay in posing the question" (translation from Japanese).

Our main hypothesis about the positive effect of embodied conversational agents with affective behavior can be divided into two specific hypotheses.

- Hypothesis 1 (*Empathy*): SC is lower when the agent shows empathy after a delay occurred than when the agent does not show empathy.
- Hypothesis 2 (*Affective feedback*): When the agent tells whether the subject's answer is right or wrong, SC is lower in the affective version than in the non-affective version.

To support Hypothesis 1 (empathy), we calculated the mean values of SC for each subject considering only the six delay game situations. Then we computed the difference between the DELAY and DELAY-RESPONSE segments on the mean values of the signal. In the non-affective version (no display of empathy), the difference is even negative (mean= −0.08). In the affective version (display of empathy), on the other hand, SC decreases when the character responds to the user (mean= 0.14). In the following, the α level is set to 0.05. The *t*-test (two-tailed, assuming unequal variances) showed a significant effect of the character's affective behavior as opposed to non-affective behavior ($t(16) = -2.47$; $p = 0.025$). This result suggests that an embodied agent expressing empathy may undo some of the frustration caused by a deficiency of the interface.

Hypothesis 2 (affective feedback) compares the means of SC values of the RESPONSE segments for both versions of the game (the agent responses of all queries are considered here). However, the *t*-test showed no significant effect ($t(16) = 1.75$; $p = 0.099$). When responding to the subject's answer, the agent's affective behavior has seemingly no major impact.

We also compared the subjects' scores in both versions. The average score in the affective version was 28.5 (from 30 answers), and 28.4 in the non-affective version. We may interpret this result in the light of the findings of van Mulken et al. (1998), who show that interface agents have no significant effect on objective measures (in their case, comprehension and recall).

In addition to taking subjects' physiological data we asked subjects to fill out a short questionnaire after they completed the quiz. The ratings are from an eleven-point scale, ranging from 0 (disagreement) to 10 (agreement). Table 1 shows the mean scores for some questions. None of the differences in rating reached the level of significance. However, the scores for the first question suggest a tendency somewhat related to the one observed by van Mulken et al. (1998), namely, that a character may influence the subjects' perception of difficulty. In their experiment though, van Mulken at al. compare "persona" vs. "no-persona" conditions rather than "affective persona" vs. "non-affective persona" conditions.

Table 1 Mean scores for some questions concerning the quiz game.

Question	*Non-affective*	*Affective*
I experienced the quiz as difficult.	7.5	5.4
I have been frustrated with the delays.	5.2	4.2
I enjoyed playing the quiz game.	6.6	7.2

Although the obtained results are still somewhat restricted, we believe that embodied conversational agents with affective behavior have the potential to alleviate user frustration similar to human interlocutors, and the assessment of user's physiological data is an adequate method to show the effects of agents.

5 Current and Future Work

We currently follow multiple parallel lines of research, which extend our work on designing and interacting with embodied conversational agents. MPML allows to script character behavior relatively easily but remains limited in creating context-sensitive, adaptive affective behavior. The SCREAM system provides a tool for the creation of sophisticated character behavior, but requires considerable effort to prepare affective responses. Authors who simply wish to include a "chatbot" to their interface might want to follow a less work-intense approach. We recently implemented an interface between MPML3.0 and a popular chatbot, the Alicebot (Mori et al. 2003). The Alicebot provides a large set of responses written in AIML (Artificial Intelligence Markup Language) that are accessible from the web. A major drawback of this approach is that agents scripted with AIML cannot easily be given a consistent personality profile and show unexpected behaviors that might be tolerable (or even desirable) for chat-style situations but not for more confined and task-specific interaction domains, such as the "Black Jack" game or educational settings. As another application, MPML has been used to implement a character-based CALL (Computer Assisted Language Learning) system that allows native speakers of

Japanese to hold English conversions with life-like agents (Juli 2003). Recently, we also started to re-implement MPML for scripting animated agents running on handheld devices. The *MPML-mobile version* allows to markup simple animations on the cellular phone platforms of two major Japanese providers.

A very promising alternative to achieve believable agent behavior is to script the environments that host the agents rather than the agents themselves. Here, Doyle's (2002) *annotated environments* concept might serve as a starting point. According to this idea, the designer of the (web) environment adds annotations to the environment that instruct the agent on how to react. Annotations might include various types of information, such as factual and affective information or the environment designer's intent. The main advantage of this approach is that agents can achieve believable behavior across various environments, and do not need any knowledge about the environment at design time.

A major focus of our current research is the use of emotion recognition technology to develop adaptive character-based interfaces (Conati 2002). We intend to process physiological data in real-time, and provide tailored agent reactions based on the user's emotional state and interaction task.

6 Conclusions

In this paper, we have described the major goals and selected results of a project on life-like embodied agents carried out at the University of Tokyo. With the aim of developing easy-to-use markup languages for synthetic characters that are capable of affective interactions with other agents – including human users – in web-based environments, we have presented the following outcomes:

- The MPML family of markup languages providing tagging structures for controlling embodied characters, presentation flow, and human-agent interaction.
- The SCREAM system offering a practical technology for specifying and scripting the mental states and processes underlying an agent's affective behavior.
- An experiment involving a character-based interface suggesting that an agent's emphatic feedback may decrease and partly undo a user's negative emotions.

In our future research, we hope to extend and refine the obtained tools and mechanisms for embodied conversational agents, in order to contribute to the vision of natural and effective interactions between humans and computers.

Acknowledgments

We would like to express our thanks to the following students for their significant contributions to the project: Takayuki Tsutsui, Yuan Zong, Peng Du, Sylvain Descamps, and Istvan Barakonyi. This research was supported by the Research Grant (1999-2003) for the Future Program ("Mirai Kaitaku") from the Japan Society for the Promotion of Science (JSPS). Sonja Mayer was supported by a internship grant from the Carl Duisberg Society (Germany).

References

André, E., T. Rist, S. van Mulken, M. Klesen, and S. Baldes. 2000. The automated design of believable dialogue for animated presentation teams. In: J. Cassell, S. Prevost, J. Sullivan, and E. Churchill, eds., *Embodied Conversational Agents*, 220-255. The MIT Press.

Arafa, Y., K. Kamyab, S. Kshirsagar, N. Magnenat-Thalmann, A. Guye-Vuillème, and D. Thalmann. 2002. Two approaches to scripting character animation. In: *Proceedings AAMAS-02 Workshop on Embodied Conversational Agents – Let's Specify and Evaluate Them!*

Barakonyi, I. and M. Ishizuka. 2001. A 3D agent with synthetic face and semiautonomous behavior for multimodal presentations. In: *Proceedings Multimedia Technology and Applications Conference (MTAC-01)*, 21-25.

BinNet Corp. 2003. Jinni 2000: A high performance Java based Prolog for agent scripting, client-server and internet programming. URL: http://www.binnetcorp.com.

Bordegoni, M., G. Faconti, S. Feiner, M.T., Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. 1997. A standard reference model for intelligent multimedia presentation systems. *Computer Standards & Interfaces* 18(6-7):477-496.

Cassell, J., H. Vilhjálmsson, and T. Bickmore. 2001. BEAT: the Behavior Expression Animation Toolkit. In: *Proceedings of SIGGRAPH-01*, 477-486.

Conati, C. 2002. Probabilistic assessment of user's emotions in educational games. *Applied Artificial Intelligence*, 16:555-575.

De Carolis, B., V. Carofiglio, M. Bilvi, and C. Pelachaud. 2002. APML, a markup language for believable behavior generation. In: *Proceedings AAMAS-02 Workshop on Embodied Conversational Agents – Let's Specify and Evaluate Them!*

Doyle, P. 2002. Believability through context. Using "knowledge in the world" to create intelligent characters. In: *Proceedings 1st International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-02)*, 342-349. ACM Press.

Du, P. and M. Ishizuka. 2001. Dynamic Web Markup Language (DWML) for generating animated web pages with character agent and time-control function. In: *Proceedings (CD-ROM) IEEE International Conference on Multimedia and Expo (ICME-01*).

Extempo Systems. 2003. URL: http://www.extempo.com.

Juli, Y. 2003. *A Computer-Assisted Learning System for English Conversation Using Animated Lifelike Agents* (in Japanese). Bachelor thesis, University of Tokyo.

Klein, J., Y. Moon, and R.W. Picard. 2002. This computer responds to user frustration: Theory, design, and results. *Interacting with Computers*, 14:119-140.

Lang, P. 1995. The emotion probe: Studies of motivation and attention. *American Psychologist*, 50(5):372-385.

Lawson, R. 1965. *Frustration: The Development of a Scientific Concept*. MacMillan, New York.

Lester, J.C., S.A. Converse, S.E. Kahler, S.T. Barlow, B.A. Stone, and R.S. Bhogal. 1997. The Persona effect: Affective impact of animated pedagogical agents. In: *Proceedings of CHI-97*, 359-366.

Marriott, A. and J. Stallo. 2002. VHML – Uncertainties and problems. A discussion. In: *Proceedings AAMAS-02 Workshop on Embodied Conversational Agents – Let's Specify and Evaluate Them!*

Microsoft 1998. *Developing for Microsoft Agent*. Microsoft Press.

Mori, J. 2003. *Affective Interaction with Anthropomorphic Agents* (in Japanese). Master's thesis, University of Tokyo.

Mori, K., A. Jatowt, and M. Ishizuka. 2003. Enhancing conversational flexibility in multimodal interactions with embodied lifelike agents. In: *Proceedings of Poster Session at International Conference on Intelligent User Interfaces (IUI-03)*, 270-272.

Murray, I.R. and J.L. Arnott. 1995. Implementation and testing of a system for producing emotion-by-rule in synthetic speech. *Speech Communication*, 16:369-390.

Okazaki, N., S. Aya, S. Saeyor, and M. Ishizuka. 2002. A Multi-modal Markup Language MPML-VR for 3D virtual space. In: *Proceedings (CD-ROM) of Workshop on Virtual Conversational Characters: Applications, Methods, and Research Challenges (in conj. With HF2002 and OZCHI2002).*

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

Ortony, A. 1991. Value and emotion. In: W. Kessen, A. Ortony, and F. Craik, eds., *Memories, thoughts, and emotions: Essays in the honor of George Mandler*, 337-353. Hillsdale, NJ: Erlbaum.

Picard, R.W. 1997. *Affective Computing*. Cambridge, MA: The MIT Press.

Prendinger, H. and M. Ishizuka. 2001. Social role awareness in animated agents. In: *Proceedings 5th International Conference on Autonomous Agents (Agents-01)*, 270-277. ACM Press.

Prendinger, H., S. Descamps, and M. Ishizuka. 2002. Scripting affective communication with life-like characters in web-based interaction systems. *Applied Artificial Intelligence*, 16:519-553.

Reeves, B. and C. Nass. 1998. *The Media Equation. How People Treat Computers, Television and New Media Like Real People and Places*. CSLI Publications. Cambridge University Press.

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

Saeyor. S. 2002. Multi-modal Presentation Markup Language Ver. 2.2a (MPML2.2a). URL: http://www.miv.t.u-tokyo.ac.jp/~santi/research/mpml2a.

Schreier, J., R. Fernandez, J. Klein, and R.W. Picard. 2002. Frustrating the user on purpose: A step toward building an affective computer. *Interacting with Computers*, 14: 93-118.

van Mulken, S., E. André, and J. Müller. 1998. The persona effect: How substantial is it? In: *People and Computers XIII* (Proceedings of HCI-98), 53-66. Berlin: Springer.