MAKING THE WEB EMOTIONAL: AUTHORING MULTIMODAL PRESENTATIONS USING A SYNTHETIC 3D AGENT

Sylvain Descamps, Istvan Barakonyi, Mitsuru Ishizuka

Dept. of Information Engineering and Communication Engineering,
School of Engineering, University of Tokyo
Email: {descamps, bara, ishizuka}@ miv.t.u-tokyo.ac.jp

ABSTRACT

Interface agents are becoming a new way for computers to communicate with humans. These agents have gained much focus recently since there is a growing interest for presentations over the Internet. The application domain of these agents is becoming wider, the quality and complexity of the existing systems is increasing fast. Our contribution to this research field concerns a new system enabling authors to easily enhance their already existing content with synthetic agents having believable behavior. It consists of a customizable 3D facial agent system and a powerful language to author presentations using interface agents, called MPML. This system provides both a versatile agent and an easy-to-use control over it.

KEYWORDS

Multimodal presentation, interface agent, facial animation, affective behavior

1. INTRODUCTION

The main goal of our research is to create presentations with animated agents, which will be mainly used on the Internet. Indeed, human presenters are still better than computer agents, however, as soon as computers perform instead of humans – e.g. distributed presentations over the Internet - the effectiveness of the presentation depends on the tools used. Traditional web pages and slides are already ways to communicate information, however, it is more natural to have a presenter who helps the understanding. The presented information's download size and time is also crucial on the Internet. We demonstrate a language called MPML (Multimodal Presentation Mark-up Language), and an agent system that provides spectacular output for users while the transmitted data over the network is kept low. We assumed that authors using our system don't have high computer skills, therefore we provided an easy-to-use yet fine control over our system.

For this project we examined research made on scriptable, animated agents. Andre *et al* (2000) made their own 2D agent system, which includes basic agent features and natural speech generation. Instead of giving precise and explicit instructions, the author chooses only the overall strategy of the presentation. It provides less control but higher autonomy for the agent. The language itself is much more complicated than ours because their system needs conditions and event structures to be defined to set-up the presentation schedule. Although 2D characters are fairly simple to use, scaling, modifying or extending their actions – not to mention creating a brand new character from scratch - usually requires considerable artistic skills. In our project we aimed at breaking this rigidity by generating real-time motion using a 3D character with parameterizable behavior.

The performed gestures of 3D agents can be categorized as facial and body gestures. Cassel *et al* (1994) made an extensive study on rule-based generation of body and facial gestures supporting speech. Gestures and locomotion of the body with scriptable animation control are discussed by Perlin *et al* (1996), which has a lot of similarities with our system. The authors indeed made a system controlling 3D agents by their own scripting language. The control they provide for the agent through the language is more powerful than ours, however, it also requires much more work from a potential author. We want our system to be used easily and to allow the author to focus on the content of his/her presentation rather than on the definition of the agent's movements. The MPML language used for authoring the multimodal content is system-independent, therefore can be used by a wide range of applications on numerous platforms. Our agent focuses on facial gestures because we want the agent to express natural and powerful emotions and personality, both of which can be realized with refined facial expressions and behavior.

2. OVERVIEW OF THE SYSTEM

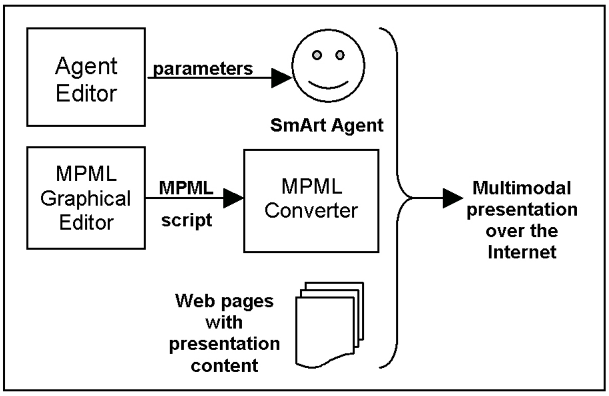
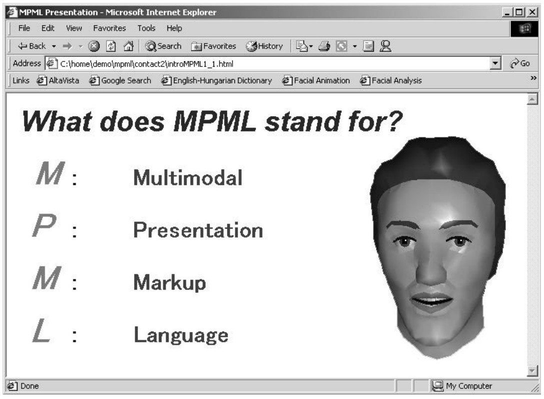


Figure 1: The flow of authoring a presentation


MPML Presentation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
History
Address
C:\home\demo\mpml\contact2\introMPML1_1.html
Go
Links
AltaVista
Google Search
English-Hungarian Dictionary
Facial Animation
Facial Analysis
Done
My Computer

What does MPML stand for?

M : Multimodal

P : Presentation

M : Markup

L : Language

Figure 2: A screenshot of our system

The procedure of authoring a presentation – as shown in Figure 1 - starts with creating the flowchart of the presentation scenario in a graphical editor. It produces the MPML script for the MPML converter, which is responsible for combining the presentation agent and the web pages used as the presentation content. These enhanced pages will control the presentation agent, the appearance and behavior of which are customized in the agent editor. An advantage of our architecture is that the language and the agent system are independent from each other, therefore the language can control several agents, while the agent can be controlled by other applications. A screenshot of the multimodal presentation as the output can be seen in Figure 2.

In our character design we didn't aim to achieve photo-realism but rather easy modification and animation, in addition to reasonable speed and quality on today's average PC platform (see Section 5). Our facial model's approach is pseudomuscle-based (Parke, Waters 1993), mainly because of limited computation time (there are speed concerns with the physics-based model, which simulates the dynamics of the skin tissue and muscle tension) and its close relation to the natural face structure. Our facial mesh and muscle data come from the IPA FaceTool project (see Harashima Lab. homepage). The novelty of our agent among other presentation agents is providing full, comprehensive control over the agent to determine its appearance and motion, and the definition of behavioral rules, which allow for automatic generation of facial gestures. An easy-to-use agent editor supports these control mechanisms. Authors are also relieved from tedious tasks like lip synchronization or taking care of elements of natural behavior like blinking or eye contact.

3. THE PRESENTER

TABLE 1: Relations made between real and synthetic presenters

Elements of presentation in real life	**Possible realization with our agent**	**Elements of presentation in real life**	**Possible realization with our agent**
The presenter's position in the room	The agent's position and size on the screen	Presentation content: slides, images, videos etc.	Same presentation content
Audience's interaction with the presenter	Mouse commands or speech input with speech recognition	Presenter's personality and appraisal to the audience	Behavior i.e. facial gestures and movements, appearance of the face
Monitoring the feedback of the audience	Facial expression recognition by a video camera (planned feature)	Presenter's mimics, facial gestures supporting speech	Synthesized facial expressions and facial colors, lip synchronization
		The tone of the speaker's voice supporting presentation material and current emotional state	Synthesized speech with varying pitch and rate parameters conveying emotional content

We wanted to add the same functionality to our presentation system as a real presenter has in a lecture room. In Table 1 you can see the relations we made. The attitude of the presenter perceived by the user is influenced by the appearance and behavior of the agent. The appearance in our case is mainly determined by the visual features of the face. In cartoons and movies positive characters most often have a charming look conforming to what most people consider attractive, while negative characters usually seem rather unappealing or have some kind of asymmetry. By changing the agent's facial mesh a wide range of faces can be created (see Figure 3). The other important character factor, the behavior has to be designed as well to create a believable agent. For this purpose we need to define behavioral primitives:

- Make a facial expression at a certain intensity
- Move the head: specific rotation, translation, zoom parameters OR turn to/away from user
- Move the eyes: specific rotation parameters for each eye OR maintain/break eye contact OR follow the mouse pointer
- Set the color of a facial part

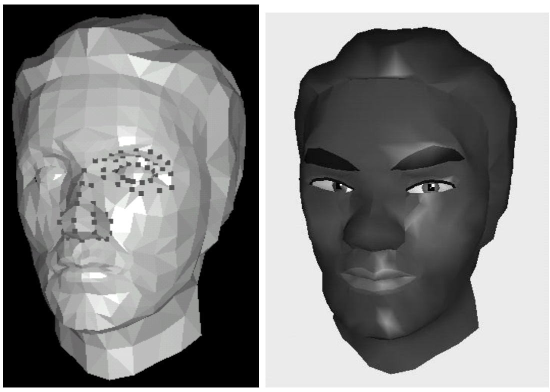
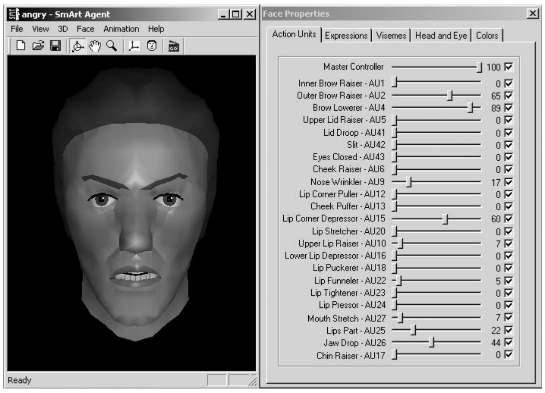


Figure 3: Facial conformation parameters



Master Controller	100	☑
Inner Brow Raiser - AU1	0	☑
Outer Brow Raiser - AU2	65	☑
Brow Lowerer - AU4	89	☑
Upper Lid Raiser - AU5	0	☑
Lid Droop - AU41	0	☑
Slit - AU42	0	☑
Eyes Closed - AU43	0	☑
Cheek Raiser - AU6	0	☑
Nose Wrinkler - AU9	17	☑
Lip Corner Puller - AU12	0	☑
Cheek Puffer - AU13	0	☑
Lip Corner Depressor - AU15	60	☑
Lip Stretcher - AU20	0	☑
Upper Lip Raiser - AU10	7	☑
Lower Lip Depressor - AU16	0	☑
Lip Puckerer - AU18	0	☑
Lip Funneler - AU22	5	☑
Lip Tightener - AU23	0	☑
Lip Pressor - AU24	0	☑
Mouth Stretch - AU27	7	☑
Lips Part - AU25	22	☑
Jaw Drop - AU26	44	☑
Chin Raiser - AU17	0	☑

Figure 4: A screenshot of the agent editor

By adding timing and duration information to these atomic components the behavior can be composed by the keyframing animation technique while taking special care for making smooth and natural transition between muscle contractions. Among these primitives, we put special emphasis on facial colors. Firstly, besides the facial shape it enables the creation of various faces with different skin, hair or iris color. Secondly, it can simulate secondary emotion effects like blushing (cheek color) or pallor (skin color). Other effects like stubble on an unshaved face (cheek and moustache color) or make-up can be also realized. An example of facial colors emphasizing facial expressions can be seen in the angry face with hot cheeks of Figure 4.

Our agent can automatically generate facial gestures. Lip synchronization is done automatically. Moreover, to provide the dynamic, realistic feature of the face, natural movements are generated by the behavior engine. The occurrence and type of these events are depending on whether the face is idle, like gazing around to avoid a static, staring face; or an action is being executed like stressing a spoken word with a head nod, raising the eyebrows at a meaningful break in the presentation or break and go back to eye contact at the end of a sentence.

4. THE PRESENTATION

4.1. Agent control

The control of the agent provides the following actions: speak, think, move and act. However, in the context of a presentation the most important action will be the speech as it will be the way the agent communicates with the user. The agent can say one or several sentences while the text is being displayed in a balloon. The parameters of the voice are set by using the current emotional state of the agent (emotion and mood). Some actions can occur at the same time as the speech when a strong emotion arises. During speech, the author can also define some special actions like making a meaningful gesture or playing a sound file, e.g. to simulate crying. The agent can also think, which means he does not speak aloud, only the text is being displayed in a balloon. This is practical when several agents would like to say something at the same time.

4.2. Affective behavior

Emotion is one important challenge for agents. Indeed, in human-human communication, emotion plays a very important role, as it adds a second meaning to the dialog, and therefore, conveys more information. The ability to control emotion would make the conversation between the user and the agent more natural.

Several models have emerged from psychology studies. One of them is the famous OCC model from Ortony, Clore and Collins (1988). The OCC model classifies 22 emotions depending on what makes them occur, and their influence on the environment of the agent. This model seems more useful when there is computation about the emergence of the emotions, like for autonomous agents with automatic emotion generation. Another theory is the “basic emotion” theory from Ekman (1992). It filters out 6 emotions (*anger*, *fear*, *sadness*, *enjoyment*, *disgust* and *surprise*). It is argued that every emotion can be expressed as a combination of these basic emotions. We referred to Ekman, Friesen (1978) and Collier (1985) when defining facial expressions for the agent concerning these 6 emotions. As for the voice, there are several works concerning the voice associated with an emotion. Murray and Arnott (1995) give some results how the voice changes according to the emotion involved. They also considered 6 emotions with only slight differences from Ekman (*anger*, *fear*, *sadness*, *happiness*, *disgust* and *grief*). We relied on their research but we had to extend to other positive emotions like *surprise* or *gratitude*, which are more often used in a presentation context than the referred negative emotions.

Our interest is more focused on the expression of emotions rather than emotion models. Indeed, the presentation author decides which emotion he/she wants to use, so we need meaningful names for emotions and then to find an accurate way to express them. To do so we have control over both the voice and the face. As for the voice, we can set the speed, pitch and volume and add some emphasis on words. We used the research of Cahn (1990) and Murray and Arnott (1995) to set these voice parameters according to the emotion, which indeed much improves the perception of the user about the emotion expression. The emotions we have chosen until now are joy, sadness, gratitude, anger, shame, surprise and fear. Besides emotion, we are also using mood. While emotion is defined as short and intense, mood is longer and has a lower intensity. It works more like a background emotional state, when no strong emotions are occurring. Please, refer to Descamps *et al* (2001) for details about the emotion and mood implementation.

4.3. Presentation control

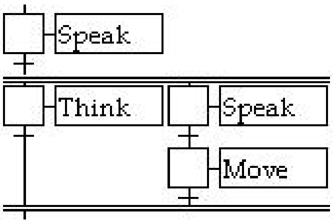


Figure 5: Parallel actions

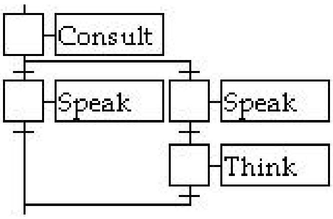


Figure 6: Branching

The organization of the presentation in MPML is mainly accomplished by a graph-like structure. A first tag, <SEQ>, defines the parts of the presentation where the actions of the agents happen sequentially. This will certainly be the case for most of the presentation. However, there is a possibility to define some parts in which actions happen simultaneously using the <PART> tag (see Figure 5 for an example output of our graphical script editor), which can become useful in case several agents are on the screen.

Another possibility concerning the structure of the presentation is a sort of equivalent of the well-known C programming language "switch" instruction. It consults the data or function given in the "target" attribute and then compares it to several values and executes the piece of script corresponding (see Figure 6). The <CONSULT> tag can either make one comparison only and then go on even if no match is found or wait until a match is found.

4.4. Interaction of MPML script with the background

In an MPML presentation we can consider that there are two different components: the agents and the background. The agents are the actors; they speak, move and thus perform a presentation. But except for the case of virtual theaters, where the presentation can be self-sufficient, the background plays a very important role as a support for the presentation, as in the case of human presentation. The common way of navigating between pages on a Web site is using *links*. However, MPML enables a new navigation method. The script can change the background page to match the explanation of the agents, using the <PAGE> tag.

Another important point concerning the background is that a large number of people already have a Web page, but only a few are using interface agent systems. It would be very unpractical for Web masters to rewrite each Web page in order to add agents to it. That's why MPML provides a way to use already existing Web pages easily. Finally, in order to make MPML not only an easy-to-use but also a powerful language, we provide a simple interface with JavaScript functions. Thus if the background contains JavaScript realizing some complex functions, it is possible to integrate and synchronize the presentation with some events happening in the Web page. This also allows the use of any content such as Flash or applet and gives a way to communicate with them from the presentation script.

The first possible interaction between MPML script and JavaScript is the <WAIT> tag. It simply consults the value of a variable or function and uses it to synchronize the presentation on events happening in the background, like user's action on buttons or the end of a movie file. It is also possible to synchronize the background on the MPML script, using the <EXECUTE> tag. This tag executes a JavaScript instruction or function and its use is limited only by the need of the author. At last, the author can introduce variable content into the speech of the agents using a variable present in the background page. It can permit some customization of the dialogue to include names or some answers from the user.

5. IMPLEMENTATION

Currently our agent is realized as an ActiveX component. Both this and the MPML converter part of the system are coded in C++, and the agent and MPML script editor software with GUI uses C++ as well with MFC classes. The rendering engine of the agent is implemented in OpenGL specification 1.1. Our application produced a 50 FPS frame rate in full screen mode in the following test environment: *Pentium III 750 MHz, 256Mb memory, NVIDIA GeForce 256 video card, Win2000 Professional, 1280x1024 screen resolution, 16-bit color depth.* Speech synthesis and recognition are realized with the Microsoft Speech API, thus all TTS engines conforming to this interface can be used with the agent.

6. CONCLUSION AND FUTURE DEVELOPMENTS

In this paper we described a system for easy creation of multimodal presentations. The content is presented by a three-dimensional facial interface agent with customizable facial expressions, behavior and voice supporting emotion and mood. Authors can easily add this agent to their previously created Web pages with the help of the graphical interface of our MPML script editor. As our system is highly customizable, the authors can add believable behavior to the agent and make the presentation even livelier. In the future we plan to enhance the multimodality of the presentation by adding facial gesture recognition. We will improve the MPML language by connecting it to a knowledge base for automatic affective control.

7. REFERENCES

Andre, E., Klesen, M., Gebhard, P., Allen, S. and Rist, T. (2000). Exploiting Models of Personality and Emotions to Control the Behavior of Animated Interactive Agents. Agents2000 Workshop.

Cahn, J. (1990). The Generation of Affect in Synthesized Speech. MIT Press.

Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Becket, T., Douville, B., Prevost, S. and Stone, M. (1994). Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. SIGGRAPH, pp. 413-420.

Collier, G. (1985). Emotional Expression. Lawrance Erlbaum Associates Inc.

Descamps., S. , Prendinger, H., Ishizuka, M. (2001). A Multimodal Presentation Mark-up Language for Enhanced Affective Presentation. ICIMADE01.

Ekman, P. (1992). An Argument for Basic Emotions. *Cognition and Emotion*, page 169-200.

Ekman, P., Friesen, W. (1978). Facial Action Coding System. Consulting Psychologists Press, Inc.

Harashima Laboratory homepage: http://www.hc.t.u-tokyo.ac.jp

Murray, I., Arnott, J. (1995). Implementation and testing of a system for producing emotion-by-rule in synthetic speech, *Speech Communication 16*.

Ortony, A., Clore, G. L. and Collins, A. (1988). The Structure of Emotions. Cambridge University Press.

Parke, F., Waters, K. (1993). Computer Facial Animation, A K Peters, Wellesley, Massachusetts

Pelachaud, C., Badler, N. I. and Steedman, M. (1996). Generating facial expressions for speech. *Cognitive Science,* **20(1):1-46**.

Perlin, K., Goldberg, A. (1996). Improv: A System for Scripting Interactive Actors in Virtual Worlds. *Computer Graphics,* **29:3**