Communicating Multimodal Information on the WWW Using a Lifelike, Animated 3D Agent

István Barakonyi, Naiwala P. Chandrasiri, Sylvain Descamps, Mitsuru Ishizuka

University of Tokyo, School of Engineering, Dept. of Information and Communication Engineering, Tokyo, Japan

Abstract

In this paper we present a 3D animated agent with synthesized face, speech and behavior, which is capable of automatic motion generation with a little user support of text annotation. Users without artistic skills are also capable of easy creation and extension of their own agents with tailor-made appearance and motion. Its implementation and compact size allows for easy use on the Internet as novel multimodal communication tools, which is demonstrated by two applications, namely the MPML and the 3D Chat project.

Keywords: facial animation, multimodal interface, lifelike animated agents

1. Introduction

This paper is a summary of 18 months of a research project done at the University of Tokyo. It is centered around the development of a synthetic three-dimensional facial agent, its concept, design questions and practical use illustrated by applications, which are the results of interdisciplinary cooperations. In this project we examined new ways of communicating information on the World Wide Web using various channels like synthetic speech, verbal and nonverbal communication, personality and traditional multimedia content with text, image, sound etc. We believe that the use of facial expressions, conversational gestures, intonated speech and visual appearance significantly enhances the often monotonous appearance of traditional web-based contents like distance education materials, computer-based presentations, chatting with humans or chatbots, storytelling with computers and so on. Firstly an overview is given on lifelike, animated agent research comparing to our chosen approaches, then our agent design concept are described followed by the introduction of two applications, finally brief implementation details and a summary conclude the paper.

2. Related work and our approach

Lifelike, animated agents with realistic behavior have gained much interest in the recent years as they seem to make dreams about having an intelligent "computer friend" come true [1]. Their function and application domain is versatile; so far they have been utilized as virtual actors [2], personal interactive tutors [3] and presentation agents [4] [5]. We visualize our agent as a three-dimensional human face (see Figure 1 and 3). Parke [6] introduced a direct parameterized model that provides control parameters for both facial conformation and expression. The action units of the Facial Action Coding System [7] and the abstract muscle action procedures of Kalra and Magnenat-Thalmann [8] both define atomic facial movements to build complex expressions as well as head and eye motion, following a pseudomuscle-based animation concept [9]. Waters [10] presents a dynamic face model with physics-based synthetic muscles and facial tissue. In our project we didn't aim to achieve photorealism but rather easy modification and smooth animation, in addition to reasonable speed and quality at today's average PC platform. The compact size and the implementation method allow for the agent's easy use on the Internet. Our facial model's approach is pseudomuscle-based, mainly because of easy editing of facial expressions, limited computation time (there are speed concerns with the physics-based model) and its close relation to the natural face structure. The face is represented by a polygon mesh, which is colored according to the facial parts. We obtained the geometric data from Harashima Laboratory's FaceTool project [11]. Multimodal user input and output [12] have been a key design issue to utilize numerous communication channels. Currently besides traditional GUI input and output channels we utilize speech synthesis and in one of the applications facial expression recognition, and consider extensions during the future development.

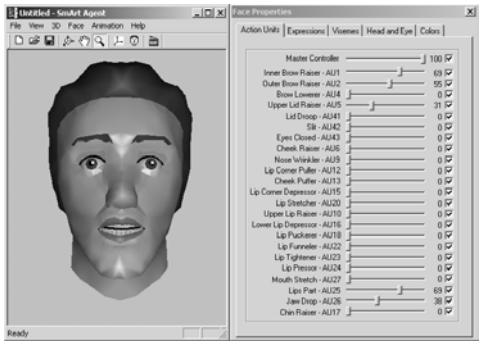

Figure 1: *Agent in the agent editor*

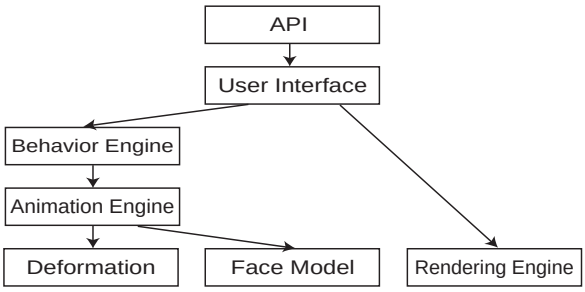

Figure 2: *Internal agent structure*

For our project we also examined research made on scriptable, animated agents. Andre et al [4] made a 2D agent system to perform multimodal presentations, 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, therefore it provides less control but higher autonomy for the agent. The language they used is much more complicated than ours described later in this article because their system needs conditions and event structures to be defined to set up the animation schedule. The performed gestures of 3D agents can be categorized as facial and body gestures. Cassell et al [13] made an extensive study on rule-based generation of body and facial gestures supporting speech utilizing linguistical analysis of the text as well. Gestures and locomotion of the body with scriptable animation control are discussed by Perlin et al [2], which has a lot of similarities with our system. The authors 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 the potential author. From the possible communication gestures our agent focuses on facial gestures. The problem with many agents currently available in the research field is the lack of user customization. Scaling, modifying or extending their behavior - not to mention the creation of a brand new character from scratch - is often not possible or requires considerable artistic skills. In our project we aimed at breaking this rigidity by generating real-time motion using an 3D character with customizable appearance and extendable behavior.

3. Structure of the agent

The synthesis and animation of human faces is a complex procedure with several abstraction levels. We use the multilayered structure shown in Figure 2, where these levels are treated independently rather than as a monolithic structure. The object at the lowest level represents the rendering engine. Besides rendering the vertices and facets of the facial mesh, it is also responsible for calculating smooth surface for the skin while preserving creases and wrinkles with a shading algorithm. The face model and the deformation component contain the geometric data of the face model and the pseudomuscles for facial expression composition. The animation engine is responsible for animating the face at an atomic level like contracting muscles or moving the eye to a certain position, and provides smooth, natural transition between these movements. The behavior engine is built on the top of these primitives and deals with higher-level units like facial expressions at a certain intensity, lip shapes for the speech or maintaining/breaking eye contact. Moreover, its task is to "bring life" into the synthetic head by automatically generating motion to support speech and give a lifelike impression like natural eye gaze, eye blinks, watering the mouth etc. The top-level object in the architecture is the API level providing a standardized communication interface towards the outside world. This is the service access point for other applications, e.g. commands that can be called from author scripts are defined here. It also handles user input like mouse or keyboard events and speech input for recognition in the future. The facial agent component can either be reused inside another application's code or scripted as an individual component.

4. Agent appearance

The user's attitude to a computer agent is influenced by its appearance and behavior. 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 facial shapes can be created (see Figure 3). Users can modify the facial parameters by fitting the mesh to a preloaded facial image in an editor, which is then converted to and saved in an appropriate format interpreted by the agent. We often say expressions like "he is red with anger" showing the importance of another significant visual feature: facial colors. We put special emphasis on it since besides the facial shape it enables the creation of various faces with different skin, hair, iris etc. color, and it can simulate secondary emotion effects like blushing (cheek color) or pallor (skin color). Other effects

Figure 3: *Creating different faces by changing facial mesh and colors*

like the stubble of an unshaved face (cheek and moustache color) or make-up can be also realized. Facial colors related to the skin (e.g. cheek, nose, eyelid etc.) are always blended with the current skin tone as its changes affect all the other facial shades. For instance, when the skin gets red, the additional redness of the cheek also gets a darker shade, and if unshaved, the stubble's bluish color becomes a bit red as well because the skin shows through.

As mentioned in the introduction, to animate the face we try to imitate the function of facial muscles or muscle groups, which are implemented as Ekman's action units. By assigning intensity values (0-100%) to these action units, we can simulate the contraction of the facial muscles at a desired degree and thus compose facial expressions. The creation of visemes - visually indistinguishable phoneme groups - for lip synchronization to the synthesized speech follows the same approach as facial expressions, because mouth shapes are also formed by facial muscles. Other controllable low-level animation parameters are eye and head position.

5. Synthesizing behavior

The other important factor how an agent appeals to us is its behavior. We build it up from atomic components that we call behavioral primitives. These are the following: making a facial expression or a mouth shape at a certain intensity, turning the head or the eyes to a certain position and changing the color of a facial part. We have to take special care of making smooth and natural transition between these facial gesture elements. Real muscles cannot contract or release immediately, they need a sort of ease-in and ease-out period. We chose the so-called Hermite function of third order (refer the work of Ostermann [14] for details), which possesses this acceleration and deceleration feature, therefore it can produce realistic motion if used as an interpolation function between muscle contraction intensity values.

By adding precise timing and duration information to the previously defined behavioral primitives we can compose higher level, complex actions with the keyframing animation technique, for example making a half smile, blushing and turning the head down at the same time can be stored and reused as an action called "ashamed". Commands like "make/break eye contact" or "turn to/away from user" are also introduced. The agent talks by using a text-to-speech (TTS) engine for synthesis. The lip synchronization is done automatically, our algorithm uses the look-ahead coarticulation model [15]. Lip shapes coming from the speech and the facial gestures are blended by an algorithm, which calculates the final action unit intensities. This algorithm is experimental and slightly biased for the facial gestures, i.e. when the character is smiling and speaking at the same time, the lip shapes showing the visemes are distorted towards a smile. There are some sounds that cannot be synthesized by the TTS engine, therefore we use sound files, e.g. to simulate crying or hiccup. An important feature of our agent is its ability to execute and precisely time all the previously defined complex actions while speaking. It is implemented by inserting tags into the speech text. For example passing the following speech string to the agent makes it act out an uneasy restaurant situation: *" **action='satisfied',120,90*\ Oh, this was a great dinner! *action='shocked',90,100*\ What? Is it really so expensive?"* The values after the action name scale the intensity and the time of the originally defined action. To provide the dynamic, realistic feature of the human 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 or raising the eyebrows at a meaningful break. We referred to [16] for background work on this topic and we are extending them with our own studies. Applications can define variables and store their values in the agent so that simple personality can be implemented. For example an external application can generate actions based on the value of the variable *friendliness* and *activity* in response to user interaction. The agent senses and is able to respond to external events; if personality is implemented, it can make appropriate reactions to them, e.g. getting surprised/angry when being bothered with the cursor.

6. Emotional voice

The agent is using a text-to-speech (TTS) engine to synthesize speech. By choosing from various TTS voices different agent representations sound in a different way representing sex, age and personality. Moreover, by inserting special tags into the text to be uttered the agent can communicate emotional content with varying intonation [17] allowing for a more natural intonation. These TTS parameters are inserted by external applications (like the MPML or the 3D Chat system described in the next section) that tailor this function to their needs. It means that each application can implement its own emotion-to-intonation parameter conversion and add these TTS parameter tags to the speech string passed to the agent in the speech command when needed. The reason why in our implementation the action tags passed to the agent don't determine the agent's intonation but the job is left to external applications utilizing the agent is demonstrated in the following examples. For instance, let's see a basic command that makes the agent speak out a sentence with no modification in the intonation:

```
Agent.Speak("How are you doing?")
```

Sometimes an application might need to add a happy tone - faster, louder and a bit higher than neutral - to the voice, for instance when the talking character is currently smiling:

```
Agent.Speak("<rate speed='0'><volume
level='90'><pitch middle='4'> Hello!
How are you doing?</pitch></volume>
</rate>")
```

However, if another application implements mood and despite the temporary smiling gesture the character is in a generally sad mood, the intonation will be affected, the parameters will shift towards a sad voice i.e. the tone becomes slower along with a lower volume and pitch:

```
Agent.Speak("<rate speed='-2'><volume
level='80'><pitch middle='0'> Hello!
How are you doing?</pitch></volume>
</rate>")
```

Therefore applications don't use a parameter mapping algorithm hard-wired in the agent but their most suitable own method.

7. Applications

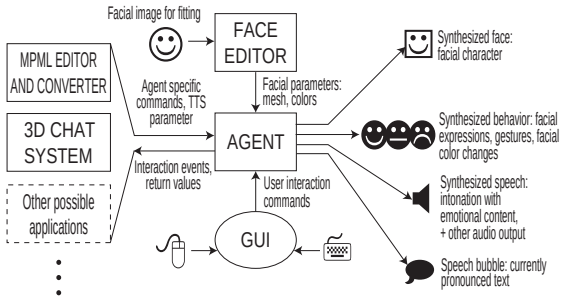

Figure 4: *Applications cooperating with the agent*

The agent has a well-defined interface towards the outside world as seen in the architecture of Figure 4. Since the agent is implemented as an ActiveX component (see the Implementation section), it can be scripted from JavaScript or Visual Basic from an ActiveX container application or reused as a C++ or VB component as an integral part of another program. External applications call these commands to have access to the services of the agent, for example speaking, performing a gesture, moving around the screen, changing the facial appearance and so on. Applications requiring emotional content in the synthesized voice are responsible themselves for inserting the appropriate TTS tags into the flow of the strings to be pronounced and sent to the agent. To create a large variety of faces for various agents the user needs to edit the facial mesh in a face editor called FaceFit [11] provided with the agent, where the mesh is fitted manually onto a facial image loaded into the background. Then the user has to choose the facial colors like skin, hair, iris, glasses etc. or use the default colors. Finally the facial model along with the colors is saved in an appropriate format suitable for the agent using a converter. This parameter file can be reused immediately as facial conformation parameters by the agent if stored on the local hard disk of the computer where we run the agent, or on the WWW at an arbitrary URL we specify. After loading these parameters the appearance of the agent instantly changes accordingly. The agent also processes GUI messages like mouse or keyboard events, which provide additional control on the agent's position, rendering parameters and behavior. The agent outputs a synthesized face with synthesized behavior and speech accompanied by a cartoon-like speech bubble for better understanding, for the hearing impaired or if TTS engine is not available/not desirable.

7.1. MPML - Multimodal presentations

One of the applications using the agent is a system featuring a language called Multimodal Presentation Markup Language (MPML) and related multimodal presentation authoring software [5]. It enables authors to easily enhance their already existing presentation content with interface agents having believable behavior. It integrates a powerful, XML-based language to author scenarios for the agents, i.e. giving instructions when, what and how to perform a certain part of the presentation. The following code snippet is part of an MPML script:

```
<scene agents="Jack,Jane">
  <page ref="page1">
    <speak agent="Jack">
      Hi Jane! How are you doing?
      <emotion assign="Jack:happy+"/>
      Haven't seen you for ages!
      <NB pause="500"/>
    </speak>
    <speak agent="Jane">
      <emotion assign="Jane:surprised"/>
        Jack, what a nice surprise!
        ...
```

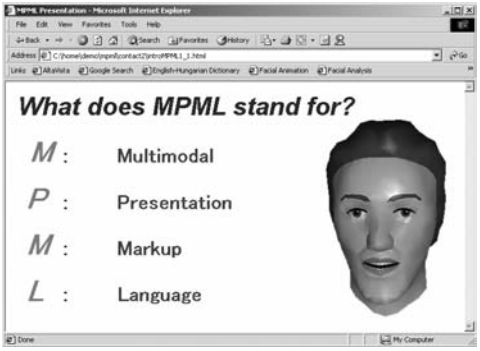

Figure 5: *Agent controlled by MPML*

Figure 6: *3D Chat system*

As the language is compliant with standard XML and independent from the agent system, it can control several different agents including our facial agent, the popular Microsoft Character Agent and other scriptable, animated interface agents. A converter software generates commands and tagged speech strings for our agent containing facial gestures information and voice intonation parameters to perform the presentation on the top of the content pages.

The procedure of authoring a presentation 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 (see the screenshot of Figure 5) will control the presentation agent, the appearance and behavior of which are customized in the agent editor. An advantage of this system is that the language and the agent system are independent from each other, therefore the language can control several agents, while the agents can be controlled by other applications. Multimodal presentations can be used in distance education materials (online and offline), storytelling, interactive conversational agent research and so on, the content created offline on the author's machine can be replayed any time online or locally on the user's machine.

7.2. 3D Chat - Chat communication enhanced by facial gestures

The other application (see Figure 6) is a chat system enhanced by real-time facial expression recognition and our animated agent. The traditional text-based chat message is accompanied by real-time facial expression/gesture information, which is sent over to the chat party on the Internet and performed and spoken aloud by the agent representing the sender. The use of the system can be demonstrated with the following example. Imagine that you type an ambiguous joke and to make yourself clear you have to insert a ":-)" (smiley symbol) at the end of the sentence. If the receiver can see the sender's real-time facial expressions synthesized by the agent, the smiley symbol can be substituted or emphasized with a smiling gesture when finished typing, so instead of character symbols you can use your natural facial gestures to communicate additional information. The sender virtually shows the agent how to behave while delivering the message.

After registering the peak facial expressions of the user in the system, various inputs are taken from the chat party: the user's current facial expression and its intensity by a head-mounted video camera and traditional GUI input like keyboard and mouse messages. Several modules process the flow of input data. The facial expression recognition module [18] calculates the MPEG-4 high-level FAP 2 (see Tekalp and Ostermann [19] for details) of the current facial expression of the user. After the recognition, special tags for the text-to-speech synthesis engine are inserted into the chat message by the emotional voice generator module to artificially generate intonation. This adds another modality to the simple text information besides facial gestures and helps emphasize the chat message's emotional content. The recognized MPEG-4 FAPs are also processed by the agent action generator, which decides on the appropriate animation command to be sent to the agent representing the user. This module chooses the type and the intensity of the animation. For example if the recognized FAP indicates "anger", depending on the intensity a short disapproving head shake or a long, furious, "red with anger" action is generated. The variety of behavioral actions and intensities prevents the agent from repetition and becoming monotonous. These agent actions are immediately replayed by the local agent representation of the user providing some feedback how he/she behaves during the chat conversation. This can generate the funny effect of seeing ourselves with the eyes of our conversational partner. The agent animation commands and the tagged chat message are then transmitted to the chat party over the Internet by the communication module. At the same time this module - as the channels are full duplex - processes data coming from other chat parties and passes it to the agents representing the chat parties to deliver their messages communicated by text, facial gestures and synthesized voice with emotional content.

8. Implementation

Currently the standalone agent is realized on Win32 platform as an ActiveX component coded in C++, and the agent editor software with GUI uses C++ as well with MFC classes. The

rendering engine is implemented in OpenGL specification 1.1. The current facial model uses 825 vertices and 1386 triangles. 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 5.0, thus all text-to-speech and speech recognition engines conforming to this interface can be used with the agent.

9. Conclusion and future developments

We designed a scriptable three-dimensional facial character agent with realistic behavior. The intention of our work is to provide an easy-to-use and extensible tool for both application developers and users, which already proved to be useful in two applications as demonstrated. As future development, experiments are considered with more detailed and sophisticated 3D face models, which call for a refined facial editor. The behavior also needs more thorough investigation, for instance by capturing personal behavior to make the agent fake the gestures of observed persons. Additional secondary emotion effects may be supported later like tears or sweating. The 3D Chat application currently lacks extensive user evaluation (except in-house testing) like measuring the acceptance of this novel system among chat users, how quickly the chat parties get used to the new modality of facial expression information (when they actually start using it intuitively) and other psychological tests.

References

1. J. Bates, "The Role of Emotion in Believable Agents," *Communications of the ACM*, vol. 37, no. 7, pp. 122–125, 1994.
2. K. Perlin and A. Goldberg, "Improv: A system for scripting interactive actors in virtual worlds," *Computer Graphics*, vol. 30, no. Annual Conference Series, pp. 205–216, 1996.
3. J. Rickel and W. L. Johnson, "STEVE: A pedagogical agent for virtual reality," in *Proc. of the 2nd International Conference on Autonomous Agents (Agents'98)*, K. P. Sycara and M. Wooldridge, Eds., New York, 9–13, 1998, pp. 332–333, ACM Press.
4. E. André, T. Rist, and J. Müller, "WebPersona: A life-like presentation agent for the world-wide web," *Knowledge-Based Systems*, vol. 11, no. 1, pp. 25–36, 1998.
5. S. Descamps, H. Prendinger, and M. Ishizuka, "A multimodal presentation mark-up language for enhanced affective presentation," in *Proc. of the Int'l Conf. on Intelligent Multimedia and Distant Education (ICIMADE-01), Advances in Educational Technologies: Multimedia, WWW and Distance Education*, 2001, pp. 9–16.
6. F. Parke, "Parameterized models for facial animation," vol. 2, no. 9, pp. 61–68, 1982.
7. P. Ekman and W. Friesen, *Facial Action Coding System*, Consulting Psychologists Press, Inc., 1978.
8. P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann, "SMILE: A multilayered facial animation system," in *Proc. of Conference of Modeling in Computer Graphics, Springer, Tokyo*, 1991, pp. 189–198.
9. F. Parke and K. Waters, *Computer Facial Animation*, A K Peters, 1996.
10. K. Waters, "A muscle model for animating three-dimensional facial expression," *Computer Graphics*, vol. 21, no. 4, pp. 17–24, 1987.
11. University of Tokyo Harashima Laboratory, "http://www.hc.t.u-tokyo.ac.jp," .
12. C. Benoit, J. Martin, C. Pelachaud, L. Schomaker, and B. Suhm, "Audio-visual and multimodal speech systems," in *Handbook of Standards and Resources for Spoken Language Systems, Supplement Volume*, D. Gibbon, Ed. 1998.
13. J. Cassell, H. Vilhjlmsson, and T. Bickmore, "BEAT: The behavior expression animation toolkit," in *Computer Graphics (Proc. of SIGGRAPH), Los Angeles, CA*, 2001.
14. J. Ostermann, "Animation of synthetic faces in MPEG-4," in *Computer Animation*, 1998, pp. 49–55.
15. M. Cohen and D. Massaro, "Modeling coarticulation in synthetic visual speech," in *Computer Animation '93*, N.M. Thalmann and D. Thalmann, Eds. Springer-Verlag, Tokyo, 1993.
16. C. Pelachaud, N. Badler, and M. Steedman, "Generating facial expressions for speech," *Cognitive Science*, vol. 20, no. 1, pp. 1–46, 1996.
17. J. Cahn, "The generation of affect in synthesized speech," *Journal of the American Voice I/O Society, MIT Press*, vol. 8, pp. 1–19, 1990.
18. N. P. Chandrasiri, T. Naemura, and H.Harashima, "Real time facial expression recognition system with applications to facial animation in MPEG-4," *IEICE Transactions INF. & SYST.*, vol. E84-D, no. 8, pp. 1007–1017, 2001.
19. M. Tekalp and J. Ostermann, "Face and 2-D mesh animation in MPEG-4," *Signal Processing: Image Communication, Special Issue on MPEG-4*, vol. 15, pp. 387–421, 2000.