Virtual Talking Heads for Tele-education Applications

Carlo Bonamico and Fabio Lavagetto

***Abstract*-- A promising technology for developing applications of tele-education concerns the use of advanced natural interfaces based on talking heads, capable of interacting with the user by means of emotional speech and facial gestures. The paper presents the latest results achieved by the IST European project INTERFACE coordinated by DIST.**

***Index Terms*—tele-education, facial animation, MPEG-4, authoring tools, advanced web programming.**

I. Introduction

TELE-LEARNING systems typically cope with the absence of a human teacher either by enriching the study material, which is then left to the student, or by complementing it with prerecorded audio/video sequences taken from real lessons. However, the bandwidth and storage limitations of networks and information systems often result in poor video quality, making the distance-learning experience less effective and more tiring for the learner. Another limitation for students is the reduced social interaction with the teacher and with other students.

A promising technology for developing tele-education applications concerns the use of advanced natural interfaces based on virtual talking heads, capable of interacting with the user by means of emotional speech and facial gestures.

These animated characters can be effectively integrated with other electronic content to act as virtual lecturers and tutors, thus enriching the lesson.

At present, character animation technologies are widely employed in movie production, although the amazing results achieved are based on the extensive use of high-end hardware and almost manual corrections and manipulations by professional animators. To create a virtual teacher that can be effectively employed in distance-learning environments, there is the need for systems that can generate animations almost automatically and display them in real time on the end-user PC, while requiring limited network bandwidth.

The development of such systems has been greatly speeded up by the recent standardization within the MPEG-4 framework.

However, many current implementations of the standard still suffer from several limitations, especially regarding model quality and web integration.

Within the IST project INTERFACE we are working towards improving user interaction with web-based applications by following a "man in the machine" metaphor. On the input side, this means that the INTERFACE software platform will include tools trying to detect the user's emotions and attitudes. On the output side, computer-animated characters will be used, acting as virtual consultants, clerks, or teachers.

This paper presents the latest results achieved by the project, with a focus on the evolution of authoring tools, as we understand that the friendliness for the production of contents is a key issue for any tele-learning platform.

The paper is structured as follows: section II discusses the characteristics of virtual characters and their role in tele-learning; section III gives an overview of the state of the art in web-based facial animation systems, while section IV presents DIST's Facial Animation Engine and related technologies. In conclusion, several directions for future research are outlined in section V.

II. Virtual Lecturers and Tutors in Tele-education

Currently, distance learning content is distributed to students mainly through a web-like architecture, either within an intranet or on the global Internet. Also many educational CD-ROMs are actually based on HTML and web technology. This happens because HTML contents can be displayed on more platforms than proprietary courseware formats.

If a facial animation system has to be chosen for implementing a virtual teacher, it is advisable to consider its limitations both in terms of network transmission and integration with other web contents.

One of the main differences between the traditional teaching methodologies and those based on distance learning is obviously the fact that the teacher and the students are not in the same place at the same time. Among the so far experimented e-education systems we can mainly distinguish three approaches based, respectively, on:

- exploiting text, audio or video-conferencing to put learners directly in touch with the tutor;
- giving students access to pre-recorded audio/video lessons;
- using only self-study material.

Apart from bandwidth limitations, the first approach is not always optimal because distance learning initiatives often aim not only at giving students access to study materials and lessons from remote locations, but also at giving them open

Manuscript received June 15, 2001. This work was supported in part by the EU project INTERFACE (http://www.ist-interface.org).

Carlo Bonamico and Fabio Lavagetto are with DIST (The Department of Informatics, System Science and Telematics) - University of Genova, Via Opera Pia 13, 16145 Genova, Italy
E-mail: {charlieb, fabio}@dist.unige.it.

access at any time, everyone at his/her own rhythm.

The use of audio/video lessons presents several limitations due to the heavy requirements in term of network bandwidth and storage.

While electronic self-study materials are usually more rich and interactive than paper-based materials, the learner is often confused by the quantity of material available, and tends to distract. The presence of a virtual guide able to present the sequence of topics and to continuously stimulate the student can help him in keeping the focus of attention on the course.

It can be difficult to implement such a guide with the streaming video of a human tutor, because it is difficult and expensive to determine *a priori* all the possible interactions between the student and the course.

In [14] two main roles are distinguished for a virtual teacher:

- lecturer;
- tutor.

The two roles share a set of common requirements, but each of them needs some more specific features. Pandzic identifies 4 basic requirements for web-based facial animation systems [11]:

- audio-visual quality;
- ease of installation;
- fast access;
- integration/interactivity.

While the visual quality of animation is important in almost any application, often the smoothness and realism of lip movement is more relevant than the complexity of the head model. This is particularly true for language teaching, where lip movements play a fundamental role in comprehension[2]. As evidenced by the literature on traditional character animation, it is possible to obtain expressiveness and liveliness even with very simple characters.

While ease of installation is a minor issue because the students have to configure the tele-learning environment only at the beginning of a course, high delays in the transmission of lessons obviously affects the naturalness of the interaction.

Integration with other web technologies is needed to synchronize the behavior of the virtual teacher with the presentation of other multimedia content: this can happen either through a scripting language like Javascript or VBScript, or through a high-level synchronization language like SMIL [23].

A. The virtual lecturer

The role of a virtual lecturer is to present one or more topics to the student. In this case, the virtual lecturer can be a valid alternative to the transmission of video lessons. Computer-generated animations are often referred to as *synthetic video* to differentiate them from *natural* content, which is acquired with a camera from the real word. Especially if the lessons have to be distributed over the Internet, the synthetic approach results in higher quality than streamed video. In fact, an MPEG-4 bitstream for a 25 frames per second animation (independently of the size of the display window) requires only 2-3 kbps. Even the most advanced streaming formats, like RealVideo or Windows Media, require at least 56kbps [17]. Also, by using appropriate server software, the animations can be even created on the fly to present new content, or to personalize the lesson in order to to target the needs of a particular student, while streaming video allows only for the reproduction of pre-recorded material. However, a high degree of expressiveness of the virtual character is required to keep the attention of the student.

B. The virtual tutor

The role of a virtual tutor is mainly that of answering to students questions and/or supervising them while they perform some task or exercise. In this case, the animated head acts as a front-end to a knowledge base that can be extended over time.

A hybrid system might use a dialog manager to answer the most common questions, and then occasionally redirect them to a human operator when no suitable answer can be found inside the knowledge base.

Implementing a tutoring system requires also the capability to monitor user behavior and react immediately to his/her inputs.

Currently, the only known animated virtual tutor is a FAQbot at Curtin University of Technology, in Western Australia, used to answer students' questions about the web-based courses [2][1].

More complex tele-education systems could integrate the two roles. Each student would be able to interact at any time with his virtual lecturer, interrupting him and asking questions (which is impossible for large number of students in the real world). In this way, each student would be offered a personalized approach to the course.

Anyway an important requirement for any multimedia system for distance education is the availability of effective authoring tools. In the case of animated virtual characters this includes both the generation of animation sequences and the customization of the face model that represents the teacher. Different characters could be used to identify different topics, or to reproduce the appearance of a particular person (e.g. having a virtual Einstein talking about relativity theory). Also, cartoon-like characters could be used inside courses aimed at children.

III. WEB-BASED FACIAL ANIMATION SYSTEMS

The earliest work with computer based facial representation was done in the early 1970's. Parke created the first three-dimensional facial animation in 1972; in 1973 Gillenson developed an interactive system to assemble and edit line drawn facial images, and in 1974 Parke proposed a parameterized three-dimensional facial model[5][7].

The 1990's have seen increasing activity in the development of facial animation techniques. At the UC Santa Cruz Perceptual Science Laboratory, Cohen has developed a visual speech synthesizer that animates a talking face starting from the text to be pronounced. The system takes into account coarticulation, that is the interaction between nearby speech

segments, in order to generate more realistic lip movements [16].

A more complete history and collection of references on facial animation and speech synthesis techniques may be found in [8] and [20].

In the last few years, a number of companies and research institutes have developed web-based animation software. They can be classified according to these four features:

- animation technology;
- rendering algorithm;
- network protocol and bandwidth requirements;
- authoring tools for content generation.

A. Animation algorithm

The main goal of real-time character animation systems is to require a minimal intervention of human animators.
This is generally achieved by splitting the generation of animations in two steps:

1. an authoring tool produces a sequence of parameters or key-frames;
2. this sequence is sent to a player that renders the deformed model.

The simplest animation algorithms are based on the concept of *key-frames*: the animator defines a sequence of expressions and then the software generates the in-between frames by using some interpolation algorithm. This approach requires that for each key-frame, the position of all the points of the face model be completely specified.

On the contrary, with parameter-based animation, only a limited set of parameters that describe the face deformations is defined for each frame. The animation software then infers the motion of all the other points of the model starting from these parameters. In turn, the interpretation of the animation parameters can be based on two different approaches:

- in the simplest case, the deformations induced by one or more parameters can be approximated with some geometric transform: as an example, the effect of a parameter describing the jaw position can be implemented trough a rotation of all the points of the model that belong to the jaw.
- more complex techniques build a physical or even an anatomic model of the human head. As an example, in the late 1980's Waters proposed a new muscle-based model, in which the animation proceeds through the dynamic simulation of deformable facial tissues, with embedded contractile muscles of facial expression rooted in a skull substructure with a hinged jaw[5][9].

After more than 3 year of standardization work, at the end of 1999, a complete set of specification for real-time facial animation systems was included in MPEG-4 version 2 [21].

MPEG-4 defines a set of animation parameters and semantic rules that can be used to control any synthetic face model compliant with the standard[18]. The main objectives of MPEG-4 specifications for the animated face object are:

- model-independent animation;
- narrow bandwidth.

The first point is important to facilitate the reuse of existing content and to be able to scale the presentation (using simpler models on less powerful hardware). The independence from the model is reached through the use of normalized animation parameters. 68 Facial Animation Parameters (FAPs) are responsible of describing the movements of the face. 66 of them specify low level actions, like the 1-D displacement of one out of 84 characteristic points (*feature points*). Two hi-level FAPs represent, with a single parameter, the most common facial expressions (joy, sadness, anger, fear, disgust, and surprise) and the 14 main *visemes*, i.e. the mouth postures correlated to a phoneme.

In order for the FAPs to be used on and extracted from any synthetic or real face, MPEG-4 introduced the use of 6 Facial Animation Parameter Units. A FAPU is the distance between some feature points (i.e. the distance between the tip of the nose and the middle of the mouth, the distance between the eyes, etc.). The value of each FAP is then expressed in terms of fractions of a FAPU. In this way, the amplitude of the movements described by the FAP is automatically adapted to the actual size/shape of the model that must be animated or from which FAPs are to be extracted.

By exploiting the high correlation between the animation parameter related to the left and right part of the face, and by encoding the parameter either with DCT or arithmetic coding, the MPEG-4 bitstream might require from 2 to 5 kbps to represent a 25 frames per second animation.

B. Rendering algorithm

Once the model representing the talking face has been deformed to assume a specific position, several techniques may be used to visualize the result[25]:

- 2D;
- 3D;
- Image-based rendering (IBR).

A cartoon-like effect is obtained with 2D techniques; in general, they are preferred when targeting less powerful hardware configurations, like palm-size PCs.

In the second case, illumination and texturing techniques are used to generate an approximation of a real head, or a completely artificial one. It should be noted that in this case the time required to render a single frame is often one order of magnitude higher than the time needed to compute the model deformations for that frame.

In the last case, instead of deforming a given 3D model, the animation is obtained by combining and deforming several base images to recreate the mouth movement due to speech, and using projection algorithms to obtain head rotations. This produces almost photo-realistic animations but limits the amplitude and variety of movements that can be reproduced[4].

Depending on the animation and rendering algorithms, the hardware requirements for an animated character can vary significantly. Several studies have evidenced that a frame rate of at least 18 fps is required to guarantee audio/video synchronization [12]. However, a current entry level PC is able to animate a medium complexity model at more than 30 fps

either with 3D Gouraud shading or IBR techniques.

C. Network transport protocol

Three different approaches are used to transmit the data that animate the virtual character from the server to the user's PC:

- download to cache;
- progressive download;
- real time streaming.

In the first scenario, the animation parameters are downloaded through an HTTP connection from a web server, and saved on the local file system. This guarantees a loss-less transfer, and reduces the complexity of the software, but at the price of introducing a noticeable delay.

Reliable transport protocols such as TCP or HTTP are used also in the second case, in which the player is also able to reproduce the animation and audio data as long as they are received from the network. In case of network congestion, however, the synchronism between audio and video might be lost, and some pauses introduced.

In the last case an unreliable protocol such as UDP or RTP over UDP is used to minimize delays and increase resilience to congestions. This approach, however, requires a more complex decoding software because it must handle packet loss and reordering.

While the bitrate required for transmitting the animation data is minimized using the MPEG-4 format, proprietary solutions typically remain in the 4-15kbps range.

The bandwidth requirement for the associated audio stream may vary from 5kbps using CELP codecs (that allows the reproduction of human voice only) to 64kbps if some background music has to be transmitted.

D. Content production

There are mainly three techniques for obtaining animations parameters and the associated audio, that differs in the amount of manual intervention required and in the capability to support different languages (Fig. 1).

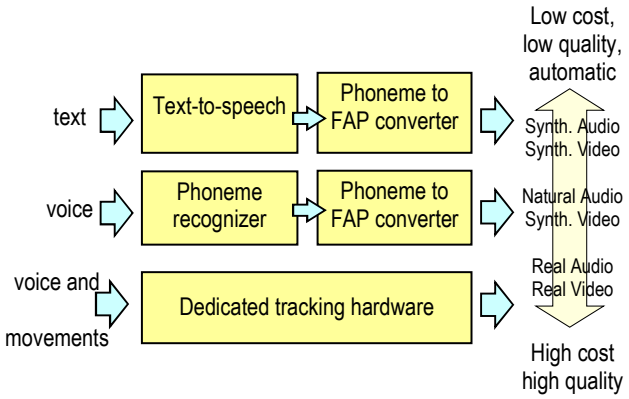

Fig. 1. The three main approaches to the generation of animation sequences, with increasing cost and animation quality.

In the first case a Text-to-Speech synthesizer (TTS) is used to create synthetic audio from plain text. Together with the audio samples, the TTS provides the sequence of pronounced phonemes and their duration. This information is the used by a phoneme-to-FAP converter to infer mouth movements corresponding to the pronounced sentence. Such a system is often referred to as a VTTS, or *Visual Text-To-Speech*.

In a second case, natural audio is used as input. By providing the pronounced text (or alternatively by processing the audio with a phoneme recognizer) the sequence of pronounced phonemes is obtained and, from it, the mouth movements, that are then synchronized with the audio stream[19].

The third solution makes use of dedicated tracking hardware, capable of capturing audio and facial movements of a real actor. The captured facial movements are encoded into animation parameter that are used to drive the virtual face. The quality of this last approach is obviously higher if compared with the other solutions, since facial expression are also captured and synthesized, while in the former cases only mouth movements are computed.

For the first two solutions, differently from the third one, which is evidently language independent, the quality of the results also depends on the specific set of phonemes used for each language.

Table 1 summarizes the characteristics of the most advanced players and authoring tools that are currently available to create animated virtual teachers on the Web.

Apart from W Interactive, Redted and CSELT that exploit a Java-based player, all the other solutions works only on the Windows platform. Some of them, like Anthropics Syntactor, works only within Microsoft Internet Explorer.

Almost all solutions exploit the standard SAPI 4.0 interface to interact with any compliant Text-to-Speech engine. Most systems however do not take into account coarticulation for the synthesis of the animations, but simply produce a concatenation of fixed mouth shapes corresponding to the pronounced phonemes. This approach does not guarantee a smooth animation of the lips.

Mendel3D's solution obtains impressive performance results even when the animated model is inserted within a more complex 3D scene. However, the authoring tool can use only motion capture or audio analysis, and the automatic generation of animations from text is not supported.

The Famous3D player can use complex head models created with a 3D professional modeling software. One limitation is that a proprietary server is required in order to host the animation sequences.

LifeFX player displays very realistic head models that are obtained through the dynamic composition of a database of images captured from real people. Because of that, only frontal views of the characters can be generated. When generating the animations with a Text-to-Speech, the user may vary the expression of the character by inserting the so-called emoticons (e.g. :-) for happiness) within the text to be pronounced.

Company and product	**Web tech. and size**	**Scripting**	**Animation and rendering (2D, 3D or Image-based)**		**Authoring tools for animations and face models Supported languages**	**MPEG4**	**URL and Notes**
Mendel3D **Mendel3DPlayer**	ActiveX or plug-in 250 kB for Windows/ MacOS/Linux	Y	geometr.	3D	**Mendel3DAvatar** : real-time speech-driven lipsinc for French/English **Mendel3D Factory**: face generator including a library of shapes, eyes, mouthes, noses that can be mixed	N	http://www.mendel3d.com
Famous Faces **Famous3D**	ActiveX or plug-in 223 kB	Y	geometr.	3D	TTS: **FamousProducer** , multilingual or motion capture **FaceAce** plug-in for 3DStudioMax for face model creation	N	http://www.famous3d.com Uses an expression mark-up language.
W Interactive **Webface**	Java Applet 150 kB	Y	geometr. [1]	2D	TTS multilingual for european languages **WebFace Workshop:** creation of face models from 2 pictures	Y	http://www.winteractive.fr
LifeFX **LifeFXPlayer**	ActiveX or plug-in 4.9MB	Y	image composer	IB	TTS or Speech-driven animation for English only	N	http://www.lifefx.com
LIPSINC Headphone and **Pulse Player**	ActiveX Plugin 370kB		geometr.	3D	Speech-driven animation	N	http://www.lipsinc.com http://www.pulse3d.com
BioVirtual **Bioplayer**	Stand-Alone 1.2 MB	N	morphing	3D	Real-time speech-driven animation **3DmeNow** for model creation from 2 pictures	N	http://www.biovirtual.com SDK available
Reallusion **CrazyTalk**	ActiveX or plug-in	Y	morphing	2D	TTS for English	N	http://www.reallusion.com/
Redted **Audiohead**	Java applet or ActiveX		morphing	3D	n/a	N	http://www.redted.com
Imagination in Motion **Realactor**	ActiveX or plug-in 260 kB		bones/skin and motion blending	3D	**RealComposer** Motion capture Model creation with 3DStudioMax	N	http://www.realactor.com Includes upper body, arm gestures.
British Telecom / Televirtual **AvTalk**	Stand-alone for Win98 5700kB		geometr.	3D	TTS for English **AvPuppet** for speech-based animation Model creation with **AvatarMe** 3D scanner	N	http://www.futuretalk.co.uk/avatars http://www.avatrme.com
Telecom Italia Lab **JOE**	VRML and Java EAI	N	geometr.	3D	TTS: **JOE Animation Studio** for Italian and Spanish Speech-driven animation : English and Italian Creation of 3D models from a single picture	Y	http://www.cselt.it/ufv/joe/
Anthropics **Synthactor**	ActiveX for Windows/IE	N	?	IB	Motion capture only	N	http://www.anthropics.com
Haptek **VirtualFriend** 3	ActiveX or plug-in	Y	morphing	3D	TTS: English, German and Spanish	N	http://www.haptek.com
Eptamedia **EptaPlayer**	ActiveX or plug-in 300K	Y	geometr.	3D	TTS: **EptaPublisher** for Italian and English Motion capture	Y	http://www.eptamedia.com

Table 1. A comparison of web-based facial animation players and authoring tools.
[1] In fact, the images of a 3D model in various poses are rendered off-line, and then sent to the client which displays them in the correct order.

The most interesting feature of British Telecom's *AvTalk* is the possibility to acquire a very realistic 3D model of a real person through the AvatarMe scanning booth. With this approach many different models can be easily created to represent various teachers for the different courses or even to show 3D representations of the students in a virtual classroom.

RealActor is a facial animation ActiveX for Internet Explorer, developed by Imagination In Motion. Its models may also have a complete body and also reproduces hand and arm gestures. However, since an authoring tool is not yet available, it seems that the only way to create animations is through motion capture.

Join Our Experience (JOE) from Telecom Italia Lab (former CSELT) achieves a very high quality of both models and animations. To display the character inside a web page, a VRML browser is required. The model is controlled by a Java applet through the External Authoring Interface (EAI).

The facial animation technology of Eptamedia (a spin-off of DIST, the Dept. of Telecommunications, Computers and System Science of the University of Genova) is described in more detail in the next section.

IV. Authoring of Distance Learning Materials using the Facial Animation Engine (FAE)

FAE has been developed by DIST through the participation to the european ACTS VIDAS project and to the activities of

the Synthetic-Natural Hybrid Coding (SNHC) group in MPEG. It represents the basic building block for applications that include a virtual character (Fig. 2).

The main component of FAE is the animation module, which takes as inputs a 3D model, an animation bitstream, an audio stream, and generates the animation on the fly, while synchronously decoding and playing the associated audio stream.

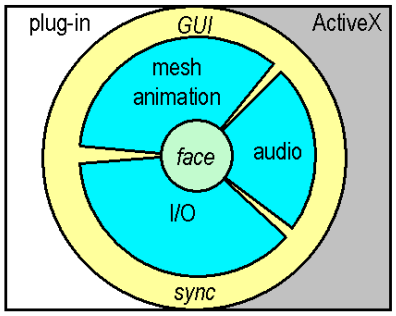

Fig. 2. Architecture of the Facial Animation Engine. The *I/O* module includes the decoders for the animation bitstream and the compressed face model format. The *Audio* module decodes and reproduces the audio stream, while providing support for synchronization with the animation. The *Mesh Animation* module is responsible for the deformation of the face model in response to the various FAP parameters. Around this core, a platform-dependent windowing and synchronization layer is built. Finally, a plug-in or ActiveX wrapping layer can be used to insert the FAE inside an HTML page and handle network connections.

As the MPEG-4 standard defines only the displacement of the 84 *feature points* as a function of the 68 FAPs, it is up to the decoding software to suitably displace any other point of the face model in order to create realistic animations. Within FAE, the position of each vertex of the model is obtained by applying several base movements (weighted translations and rotations) to the vertices surrounding each feature point. FAE is able to animate different models, because these basic deformations are not hard-wired, but are loaded from a model-specific *semantic file*.

FAE is almost entirely written in ANSI C, and exploits the multi-platform OpenGL library for rendering. Only the audio module and the windowing layer are platform-dependent.

The software is able to display a fluid animation, at more than 20 frames per second with a medium complexity model (2000 polygons) on a rather obsolete Pentium II machine without an accelerated video card. If 3D hardware acceleration is available, models up to 15000 polygons may be animated at 25 fps. Anyway, an artist can create effective 3D models with as few as 1500 polygons.

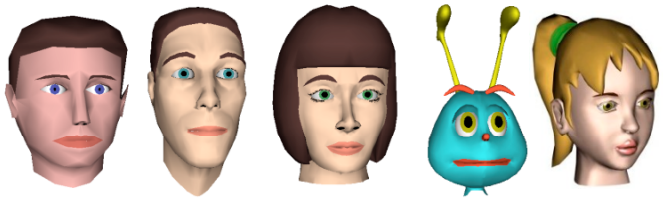

Fig. 3. Some MPEG-4 compliant face models used by the Facial Animation Engine. Complexity ranges from 750 polygons for model Mike (left) to 5000 polygons for model Asia (right).

With the use of a CELP audio codec and of the standard MPEG-4 codec for transmitting the Facial Animation Parameters, the overall bandwidth requirements remain under 8 kbit/s, which means that FAE could be effectively used even over an analog modem or a GPRS network.

Around the FAE core, both a plug-in for Netscape and an ActiveX control for Internet Explorer have been built. Thus, the virtual character can be integrated in HTML pages and controlled through Javascript. The scripting interface allows for:

- changing the model, even on the fly during an animation;
- displaying a background image;
- loading an animation stream from a remote URL;
- stopping and replaying the current animation.

The player can also generate some events to synchronize the presentation of other multimedia objects with the end of animation, as an example, or to concatenate several streams.

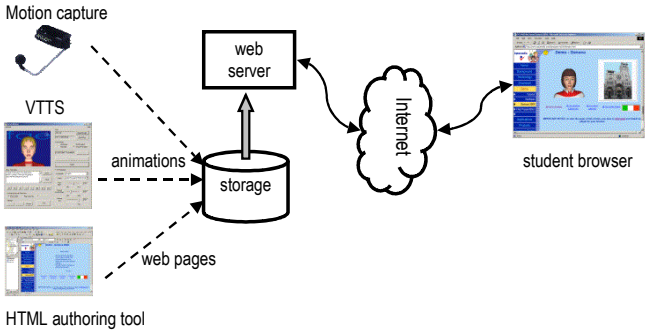

Fig. 4. Authoring and deployment of web-based courses with FAE. A dedicated server is not required, as the animation and audio streams can be stored on any web server.

The animation sequences can be produced either by a text-to-animation module or with motion capture. Currently, the generation of lip movements from audio is not yet implemented.

In the first case, an authoring tool interfaced with a generic SAPI-compliant speech synthesizer is used to produce automatically both the synthetic voice and the corresponding face movements from plain text. This approach is the most effective when the content has to be updated frequently, or when a speaker for a given language is not available.

The generation of the animation parameters is performed by the phoneme-to-FAP converter module (Fig. 5). Starting from the phonetic sequence and timing information provided by the TTS module, the animation stream is built by extracting the

visemes corresponding to each phoneme from a database and by concatenating them together. To take into account the coarticulation phenomenon and produce more realistic animations, a different representation of the visemes is used depending on the preceding and following phonemes. The FAP trajectories are then resampled according to the timing information, and finally filtered to obtain a smoother animation.

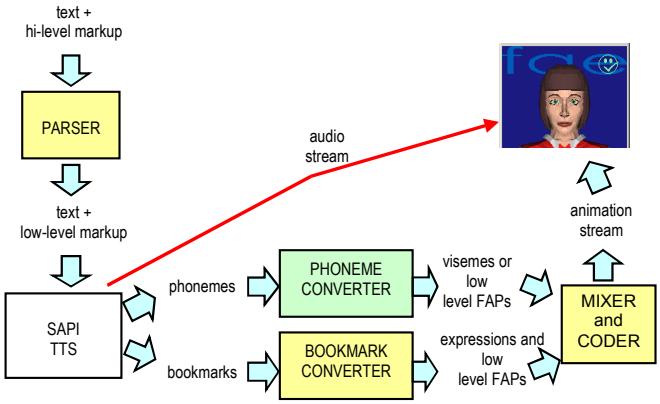

Fig. 5. Synthesis of animation sequences within the EptaPublisher authoring tool. Starting from plain text, optionally tagged with high-level information (e.g. "<happy>good morning, how are you?"), a parser produces the low-level markers that are recognized by the TTS. The phonetic sequence and timing information produced by the TTS is used to compute the Facial Animation Parameters that represent both the lip movements and the face expression.

To build the database, a quantity of speech with the associated mouth movements has been acquired using a 3D tracking hardware (the Elite system at Polytechnic of Milan). This corpus has then been manually segmented to identify the trajectories of the 8 animation parameters that control the shape of the mouth during speech production.

While the general algorithm is language-independent, the animation quality can be improved by using language-specific viseme databases. At present, the system has been fine-tuned for Italian and English.

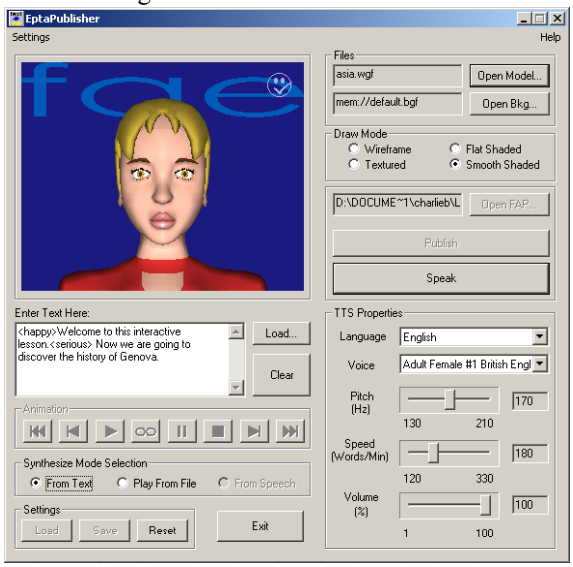

Fig. 6. The graphical interface for the authoring of animation sequences. It allows for the customization of the characteristics of the synthetic voice. It is possible to preview the created animations with different face models.

EptaPublisher is a graphical front-end to this module (Fig. 6). It also adds support for a high-level tagging scheme, which allows for the control of the prosody of the generated animations. As an example, an high level tag (e.g. <*SAD*>) is mapped to two different low-level entities: some audio parameters that alter the synthetic voice (e.g. a low pitch of the voice in the case of sadness), and some bookmarks that are passed to the FAP-generator. These can indicate either an expression between the 6 defined by the MPEG-4 standard (joy, anger, sadness, surprise, fear, and disgust) or the value of a single FAP (e.g. changing the FAP that controls the head rotation around the Y-axis to obtain a nodding).

The publisher integrates in a single application also the audio encoder and the FAP encoder. In this way, the content can be directly uploaded to the web server

To let the virtual teacher acts as an alter-ego of a real teacher, a motion capture system can be used to record his real voice and mimics. A FAP encoder that uses dedicated tracking equipment to acquire facial movements in real time has been developed. The X-Ist tracking system is basically a lightweight helmet with an infrared camera. It is able to track the 2D motion of small reflective markers on the speaker face. A gyroscopic sensor is attached on the back of the helmet to record head rotations. This technique is the most suitable when longer lessons must be recorded, and the limited prosody of the speech synthesizer becomes an obstacle in keeping the student attention.

Finally, the customization of face models is possible trough the VisualEditor, a graphical interface built with Java3D[10]. This application can import face models in VRML2 format, and lets the animator add the specific information needed to animate them within the FAE by simply selecting and labeling vertices. Optionally, the software can perform a rough automatic labeling which can then be manually refined.

The models are then exported either in VRML2 format or in a proprietary compressed format. Each compressed model takes from 30 to 100kB.

V. Directions for Future Research and Conclusions

An important issue is making the virtual teacher more autonomous, by integrating it with an A.I. engine, like an intelligent agent or a dialog manager. This would let the tutor answer arbitrarily phrased questions from the user, and would also permit the personalization of the lesson according to the profile of each student.

One obstacle to the diffusion of virtual lecturers is the limited prosody of currently available speech synthesizers. Even the ones that produce a very realistic-sounding voice are anyway quite monotone. Inside the INTERFACE project, a partner is working on an emotional speech synthesizer. We are also studying the interaction between the speech synthesis and the facial animation engine so that a change in the prosody of the voice is accompanied by a corresponding change in the expression of the virtual face (the so-called visual prosody).

Following the approach successfully taken by the STEVE

project [13], where a virtual trainer uses his whole body to show how to perform various tasks needed to control the engines of a ship, the addition of the body to the virtual teacher is under consideration. Especially when the virtual tutor is supervising some exercise undertaken by the student, providing a feedback based on body gestures can be less obtrusive and more effective than a feedback based on speech or text [14]. As an example, the virtual tutor could nod in sign of approval, or shake the head, to disapprove if the user is making some mistakes.

On the long term, we can image the creation of a 3D virtual classroom where not only the tutor is represented by an avatar, but also other students that are accessing the system at the same time. The awareness of the virtual presence of other classmates would improve the didactic interaction between the students, and give the possibility to do more complex exercises.

In conclusion, parameter-based facial animation is now a mature technology. In fact, it is now possible to animate a 3D virtual character in real time on a conventional PC. This technology can thus be effectively used to implement virtual lecturers and tutors for distance learning applications.

However, there is still room for many improvements, especially concerning the reproduction of the expressiveness of a human teacher and on the improvement of the knowledge-based systems that control the virtual teacher. Maybe one of the most difficult challenges in this field is making the teacher also engaging: the best teachers not only are precise and effective in explaining a topic, but are able to fascinate their students.

ACKNOWLEDGMENT

The authors would like to acknowledge Maurizio Costa and Roberto Pockaj for their valuable suggestions.

REFERENCES

[1] A. Marriott "A Java based Mentor System", to appear in *Java in the Computer Science Curriculum,* T. Greening Ed., LNCS, Springer-Verlag

[2] S. Beard, B. Crossman, P. Cechner, and A. Marriott, **"**FAQBot" in. *Proc. of Pan Sydney Area Workshop on Visual Information Processing*. Sydney, Australia, November 10, 1999.

[3] D. W. Massaro *Perceiving talking faces: from speech perception to a behavioral principle,* Mit Press, December 1997

[4] E. Cosatto and H. Graf, "Sample-based synthesis of photo-realistic talking heads", *Proc. Computer Animation '98*, Philadelphia, USA, pp. 103–110

[5] F. I. Parke and K. Waters, *Computer Facial Animation*, AK Peters, 1994.

[6] F. Lavagetto and R. Pockaj, "The facial animation engine: towards a high-level interface for the design of MPEG-4 compliant animated faces", in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, n. 2, pp. 277-289, March 1999.

[7] F. Parke, "Parametrized Models for Facial Animation", in *IEEE Computer Graphics Applications*, vol. 2, n. 9, pp.61-68, November. 1982.

[8] *Facial animation reference list,* http://mambo.ucsc.edu/psl/fan.html.

[9] J. Fischl, B. Miller, J. Robinson "Parameter tracking in a muscle–based analysis-synthesis coding system", in *Proc of Picture Coding Symposium*, Lausanne, Switzerland, March 1993.

[10] H. Sowizral, K. Rushforth, and M. Deering, *The Java 3D API Specification,* Adison-Wesley, 1998.

[11] I. Pandzic, "Life on the web", to appear in *Software Focus,* John Wiley & Sons, 2001

[12] I. Pandzic, J. Ostermann, and D. Millen "User evaluation: synthetic talking faces for interactive services", in *The Visual Computer Journal*, vol. 15, n. 7-8, pp.330-40, Springer-Verlag, 1999

[13] J. Rickel and W. L. Johnson "Task-oriented collaboration with embodied agents" in J. Cassell, J. Sullivan, and S. Prevost Eds., *Embodied Conversational Agents*, MIT Press, 2000.

[14] W.L. Johnson, "Pedagogical Agents", to appear in the *Italian AI Society Magazine*.

[15] M. Cohen, and D. Massaro "Development and experimentation with synthetic visual speech", *Behavioral Research Methods, Instrumentation, and Computers*, n. 26, pp. 260-265, 1994

[16] M. Cohen, and D. W. Massaro "Modeling coarticulation in synthetic visual speech", in N. Thalmann, and D. Thalmann Eds. *Computer Animation*. Springer, Tokyo, 1993

[17] M. Crudele, "Delivering video teaching courses through the Internet: technical issues", in *Procedings of the Aica 98 Conference*, Napoli, 18-20 November 1998

[18] P. Doenges, T. Capin, F. Lavagetto, J. Ostermann, and I. Pandzic, "MPEG-4: Audio,Video and synthetic graphics for mixed media", in *Image Communications Journal*, special issue on MPEG-4, vol. 9 n.4 pp. 433–463, 1997

[19] P. Eisert, S. Chaudhuri, and B. Girod "Speech driven synthesis of talking head sequences", in *3D Image Analysis and Synthesis*., pp 51–56, Erlangen 1997.

[20] P. Rubin, E. Vatikiotis-Bateson, "Talking heads", in D. Burnham, J. Robert-Ribes, and E. Vatikiotis-Bateson, Eds.: *International Conference on Auditory-Visual Speech Processing-AVSP 98*, Terrigal, Australia, 1998,pp. 231-235.

[21] R. Koenen, "MPEG-4 Multimedia for our time", in *IEEE Spectrum*, vol. 36, n. 2, pp. 26-33, February. 1999.

[22] I. Pandzic, J. Ostermann, and D. Millen "User evaluation: synthetic talking faces for interactive services", in *The Visual Computer vol.* 15 n. 7/8, pp 330-340, 1999.

[23] W3C consortium, "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification**",** W3C Recommendation 15-June-1998 http://www.w3.org/TR/REC-smil

[24] Y. Bodain and J. Robert, "Investigating distance learning on the Internet", in *Proc. Of Inet2000*, Japan, 18-21 July 2000, http://www.isoc.org/inet2000/cdproceedings/6a/6a_4.htm

[25] T. Möller and E. Haines *Real-Time Rendering*, A.K. Peters, 1997.

Carlo Bonamico was born in Genoa, Italy, on February 27, 1974. He obtained his "laurea" degree in electronics engineering from the University of Genova in 1998. Since 1999, he is a PhD student at DIST's DSP Lab. His research interests include facial animation and streaming of multimedia data over the Internet. He is actively involved in the IST projects INTERFACE and Origami.

Fabio Lavagetto was born in Genoa, Italy, on August 6, 1962. He received the "laurea" degree in electrical engineering from the University of Genoa, Italy, in March 1987. From 1987 to 1988 he worked by the Marconi Group on real-time image processing. In November 1988 he joined DIST,the Department of Communication, Computer and System Sciences, University of Genoa, receiving the PhD degree in 1992. He was visiting researcher by AT&T Bell Laboratories, Holmdel, NJ. At present he is Associate Professor at DIST, University of Genoa where he teaches a course on Radio Communication Systems. In 1995-1999 he coordinated the European ACTS project VIDAS, concerned with the application of MPEG-4 technologies in multimedia telecommunication products. Since January 2000, he coordinates the IST European project INTERFACE oriented to speech/image emotional analysis/synthesis. He is the author of more than 60 scientific papers in the area of multimedia data management and coding.