MAge-AniM: a system for visual modeling of embodied agent animations and their replay on mobile devices

Luca Chittaro
HCI Lab, University of Udine
via delle Scienze, 206
33100 Udine, Italy
chittaro@dimi.uniud.it

Fabio Buttussi
HCI Lab, University of Udine
via delle Scienze, 206
33100 Udine, Italy
buttussi@dimi.uniud.it

Daniele Nadalutti
HCI Lab, University of Udine
via delle Scienze, 206
33100 Udine, Italy
nadalutti@dimi.uniud.it

ABSTRACT
Embodied agents are employed in several applications (e.g. computer-based presentations, help systems, e-learning and training, sign language communication for the deaf), but the process of developing them is still complex (e.g., modeling animations is one of the difficult and time-consuming tasks). Moreover, although mobile devices have recently reached a performance level that allows them to manage 3D graphics, most embodied agents run on desktop computers only. The aim of our research is twofold: (i) proposing a tool that allows novice users to approach the animation modeling process of 3D anthropomorphic agents in a simple way, and (ii) proposing a 3D player to display these animated agents on PDAs. Besides discussing in detail the proposed system, the paper reports about its informal evaluation and two of its applications: sign language animation for the deaf and mobile fitness training.

Categories and Subject Descriptors
H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *Artificial, augmented, and virtual realities*; I.3.6 [**Computer Graphics**]: Methodology and Techniques – *Interaction techniques*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism – *Animation*.

General Terms
Human Factors.

Keywords
Visual Animation Modeling, Embodied Agents, Mobile Devices.

1. INTRODUCTION
In recent years, several research projects have employed embodied agents in applications such as computer-based presentations [3][29], e-commerce websites [22] and learning environments [25]. The adoption of embodied agents is motivated by several considerations: they help to lower the "getting started" barrier by allowing for the emulation of human-human communication [1], since the body is well equipped to support conversation [13]; they make computer-based presentations more lively and appealing, because they can attract user's attention and they can convey additional conversational and emotional cues [3]; they have a positive impact on factors such as likeability, engagement, and perceived intelligence in e-commerce applications [22]; they can have a positive effect on a student's perception of her learning experience [25].

Sign languages are a particular application domain for embodied agents: embodied agents can teach sign languages [23][31] and translate text into sign language animations [38]. Using embodied agents instead of videos in sign languages presents several advantages. With videos, one has usually to associate a video to each single word in sign language, so a sentence corresponds to a concatenation of videos, with unnatural gaps between subsequent videos. With embodied agents one can obtain smooth transitions and make the resulting animated sentence more realistic; besides, since agent animations require less space and bandwidth than videos, a wider range of sign language applications becomes possible.

A very recent development in embodied agent research concerns mobile devices. The first experiments reported in the literature are about applications such as collaborative virtual environments [17] and health advisors [8], but embodied agents on mobile devices are likely to improve also existing mobile applications: for example, location-aware mobile tourist guides [6] may be enhanced by introducing an embodied agent that interacts with the user as a human tourist guide. Unfortunately, most developed embodied agents still run on desktop computers only.

Since modeling embodied agent animations is a time-consuming and difficult task, research (e.g. [11][15][19]) as well as commercial (e.g. [2][4]) solutions to support animators have been proposed. These solutions tend to focus on a particular phase of the animation process (e.g. body modeling, low-level or high-level animation modeling) or are designed for a specific context (e.g. dialogue animation, sign language animation) or for particular target users (e.g. animation experts, 2D animators). In general, people without a background in computer animation may encounter difficulties in using existing tools.

The aim of our research is twofold: (i) proposing a tool that allows novice users to approach the animation modeling process of 3D anthropomorphic agents in a simple way, and (ii) proposing a 3D player to display these animated agents on PDAs. This paper proposes Mobile Agent Animation Modeler (MAge-AniM), a system that allows for simple animation modeling, based on a familiar metaphor and immediate visual feedback. Moreover, MAge-AniM guarantees the portability of the modeled animations, which can be played both on mobile devices (through its player) and desktop systems.

The paper is organized as follows. Section 2 surveys related work. Section 3 presents the MAge-AniM system. Section 4 describes two applications where MAge-AniM has been employed. Section 5 discusses the results of an informal evaluation. Section 6 provides conclusions and outlines future research directions.

2. RELATED WORK

2.1 Embodied Agents

Some general architectures for embodied agents have been proposed in the literature, e.g. Rea [12] and MIAU [28] are two well-known examples.

Rea [12] is an architecture for an anthropomorphic conversational agent that uses body language, eye gaze and facial expressions to organize and regulate the conversation. Cassel et al. [13] define conversational agents as those that have the same properties as humans in face-to-face conversation. In particular, they have the ability to: recognize and respond to verbal and nonverbal input; generate verbal and nonverbal output; deal with conversational functions such as turn taking, feedback, and repair mechanisms; give signals that indicate the state of the conversation; contribute new propositions to the discourse [13]. Rea has been designed to respond to visual, audio and speech cues normally used in face to face conversation and to generate these cues when it talks.

MIAU [28] is a platform that can be used to build a wide range of embodied agent applications with different conversational settings. MIAU clearly distinguishes between an agent's “brain” and its “body” by separating behavior planning and agent player components. MIAU integrates an architecture for life-like embodied agents, called PPP (Personalized Plan-based Presenter) Persona [3][29]. A PPP Persona agent is able to show, explain, and verbally comment textual and graphical output on a window-based interface. Besides performing tasks requested by users, the agent has an independent basic behavior for idle-time animations (when the user does not interact with the interface) and reacting animations (when the user begins interaction).

The importance of evaluating conversational agents is underlined in [30]. Besides providing guidelines for different evaluation methodologies and for design and evaluation parameters, the book describes some experiences with conversational agents in different applications (e.g., education and e-commerce).

2.2 Mobile Embodied Agents

Mobile devices have recently reached a performance that allows them to manage 3D graphics, so using embodied agents on these devices has become possible. Bickmore [7] presents a study of users having conversations with animated embodied agents on PDAs. This study demonstrates that, in conversations with embodied agents, most people use the same nonverbal behavior that they use in normal everyday conversation, although frequency is lower in conversations with embodied agents. The study also demonstrates that the reduced size of mobile agents and the fact that users hold them in their hands do not have significant impact on the interaction.

Mobile embodied agents have been used in the literature to play different roles such as personal assistants, personal friends or motivators. Moreover, mobile embodied agents have been used for real-time controlling avatars in projected multi-user environments.

One of the first conversational agents for mobile devices is described in [21]. The application runs on a PDA and provides various personal services, such as appointments, e-mail and weather forecast, using speech recognition and synthesis, and an embodied agent with facial animation to interact with the user. However, to overcome PDA limitations, processing is carried out on a remote server: the client on the PDA streams audio to the server and plays back audio and facial animation scripts sent by the server as feedback. The scripts contain a sequence of mouth positions and durations of each position. There are 18 possible mouth positions, each one associated with an image. When the client receives a facial animation script and an audio file, it plays them in sync.

Gutiérrez et al. [17] used an embodied agent on a PDA as interface for real-time control of avatars in multi-user visual environments to eliminate the display of menus or other widgets on the simulation screen of the multi-user environment. Mobile users are provided with a simplified version of their own avatar on the PDA to modify its posture and location on the simulation screen.

The possibility of using mobile embodied agents in health behavior change (e.g. personal trainers in fitness exercise or motivators to quit smoking) and relationship building (e.g. “buddies” which help autistic users finding new friends, after learning social interactions by interacting with the agents) is discussed in [8].

2.3 Tools for modeling embodied agent animations

Since modeling animations for embodied agents is difficult and time-consuming, several researchers [11][15][19][37] proposed visual tools to simplify and speed-up this task. The proposed solutions differ in the phases of the modeling process they focus on, in the employed techniques, in the intended users or contexts they are meant for. For example, Carretero et al. [11] proposed a tool that is specifically designed for dialogs and other animations where speech is predominant. The tool focuses on high-level animation features, such as synchronization between speech and gestures and behavioral animation (e.g. the user can affect animations by choosing if the embodied agent will be happy, angry, anxious, etc.). The visual interface of the tool lets the user choose one or more embodied agents and then type the text they have to read. The user can select the embodied agent behavior by changing the color of the text and insert animations by dragging special icons between words. The tool is an interesting solution for dialog animation, but cannot be efficiently used for other kinds of animations, because the predefined animations and the tool interface are very context-specific.

The tool proposed in [15] is specifically aimed at a particular category of users, i.e. 2D animators. It considers low-level aspects of agent animation modeling. Since an animation is subdivided in a sequence of postures (i.e. a set of rotation values for the joints of the embodied agent skeleton), the visual interface exploits the intended users' ability in 2D animation, allowing them to draw 2D sketches of the postures the embodied agent must take. For each 2D sketch, the tool computes all the possible 3D postures which can be mapped into the sketch and then, considering the entire sequence of sketches, a 3D animation is proposed. If the animation is not the desired one, users can edit it, by changing the selected 3D posture for one or more sketches.

A more general visual modeling tool is proposed in [19]. The tool introduces the idea of animation reuse, using a database from which animations and 3D models can be retrieved. All animations can be visually and interactively adjusted; e.g., the user can change a female walking animation into a male one, or change the destination an embodied agent has to reach.

An approach based on reuse is proposed also by [37] for a visual tool to create animations of sign language gestures (e.g. hand, arm, head movements) and signs (i.e. sequences of gestures corresponding to a word of the language). Since the same gesture can be reused in several signs and new signs can be obtained by composing existing ones, the modeled gestures and signs are stored in a database for future reuse. The stated final goal for the tool is to allow deaf people to "write" in their own language, composing sentences using the stored signs.

Alice [14] is a visual tool that helps the user in modeling simple 3D animations. Although this tool is not meant for embodied agent animation, it inspired our work because it is one of the few animation tools that focuses on novice users. Besides, Alice does not require specific 3D graphics and mathematics background (e.g. matrices, radians) and the user can easily preview the animations while she builds them. It can be used also by kids without any programming experience that use their computers only for common tasks, such as word processing or web surfing.

3. MAGE-ANIM

To simplify the modeling of embodied agent animations and display them on mobile devices, we propose MAge-AniM, a system composed by two main parts (Figure 1): (i) the *H-Animator* tool supports the user in each phase of the animation modeling process, and (ii) the *MobiX3D* player allows the user to view embodied agent animations on a mobile device.

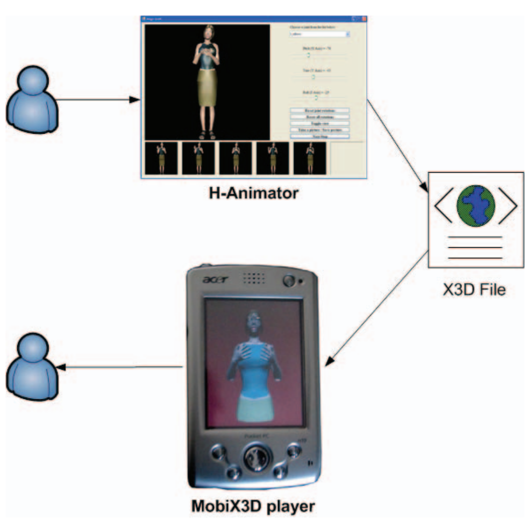

Figure 1. High-level diagram of MAge-AniM.

3.1 H-Animator

As shown by [19][37], composing long animations is easier and faster if the animator can reuse short animation sequences that have been previously stored. Therefore, in MAge-AniM, animations are organized in two classes: (i) *simple animations* are short animations, such as a single action (e.g. pointing, jumping, kicking) or a simple gesture (e.g. moving the hands to emphasize a spoken word or to represent a word in a sign language), and (ii) *complex animations* are sequences of simple animations.

H-Animator uses a direct kinematics per-key approach for simple animations. This approach subdivides an animation into a sequence of keyframes at which the embodied agent assumes the main postures of the animation. For each posture, the user has to set the rotation values that should be applied to the agent joints and specify the time at which the posture should be taken. Direct kinematics per-key animation modeling can be divided into two sub-tasks [33]: the specification of keyframe values and the keyframe timing. Terra and Metoyer [33] proved, studying novice users, that the specification of keyframe timing is more difficult than the specification of keyframe values and a clear separation of these two sub-tasks leads to an easier modeling process. Therefore, H-Animator separates animation modeling into three distinct phases: the two phases identified by Terra and Metoyer (*posing* and *timing*), and a *joining* phase. Users repeat the posing and timing phases to model simple animations, which can be reused in the joining phase to build complex ones.

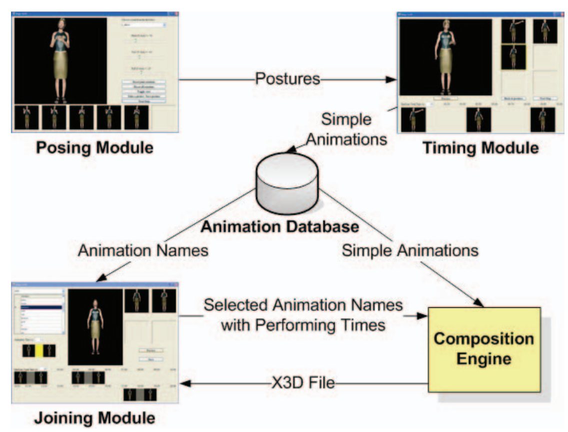

Figure 2. H-Animator architecture.

Figure 2 shows the architecture of H-Animator. The *Posing Module* handles the posing phase, where the user visually models the embodied agent postures. Postures are sent to the *Timing Module*, which is responsible for the timing phase, where the user specifies the time at which the modeled postures have to be assumed. As a result, the Timing Module generates a simple animation, which is stored in the *Animation Database.* The *Joining Module* implements the joining phase, where the user composes the simple animations to build a complex one. The Joining Module retrieves the names of the available simple animations from the database and lets the user choose which ones she wants to use for a complex animation. A list of selected animations and the times at which they should be performed is sent to the *Composition Engine*, which retrieves them from the Animation Database and generates the transitions between each simple animation and the next one. The output of this module is a file that specifies the complex animation, following the X3D [35] and H-Anim [20] standards, respectively for portable 3D content and interchangeable humanoid animation. The resulting animation can be previewed using the Joining module, and played on X3D browsers, such as BS Contact [9]. The implementation details of H-Animator are technically discussed in [10].

3.1.1 Posing

To simplify the posing phase, we chose a photographer's metaphor: the user can pose the embodied agent and then she can "take pictures" of it. Figure 3 shows the H-Animator interface in this phase: on the left, a wide interactive window allows the user to view the agent from different viewpoints; on the right, a set of controls allows the user to change the agent posture. To select the joint that has to be rotated, the user can choose its name in a combo-box or directly click on it in the interactive window that displays the agent.

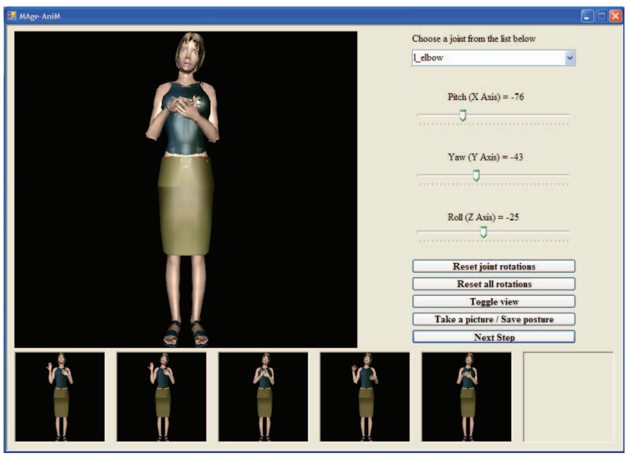

Figure 3. H-Animator interface for posing.

The rotation that has to be applied to the selected joint can be changed using three sliders ("Pitch", "Yaw" and "Roll"). Each slider allows one to change a component of the angle in the Eulerian notation and displays the rotation value in degrees. In the X3D ISO standard, adopted by the MAge-AniM system, rotation values have to be specified in the axis-angle notation using radians, but this notation is usually unfamiliar to people without a specific mathematics background, so H-Animator performs the trivial conversion between degrees and radians as well as the conversion between the axis-angle and the Eulerian notation. Therefore, as the user interacts with the sliders, the rotation values are immediately converted and constantly applied to the chosen joint of the embodied agent in the interactive window, giving feedback about the posture the agent will take. So the user can pose the agent in real-time and then, when the desired posture is obtained, she can "take a picture" of it. All the pictures taken are displayed on the lower area of the interface and are automatically associated to a data structure containing the rotation values; whenever the user clicks on a previous picture, the corresponding rotations can be thus loaded and applied to the embodied agent joints in the interactive window.

3.1.2 Timing

After collecting the desired pictures, the user can switch to the timing phase. To be consistent with the photographers' metaphor, this phase consists in sorting the pictures. The task is extremely easy: the collected pictures are displayed on the upper-right side of the H-Animator interface (Figure 4); when the user clicks on them, the corresponding posture is immediately taken by the embodied agent in the interactive window on the left; after visually checking the posture, the user can click the timeline in the lower part of the interface at the instant where that posture has to be placed.

Figure 4. H-Animator interface for timing.

3.1.3 Joining

All the simple animations modeled in the timing phase are automatically stored in the Animation Database. In the joining phase, H-Animator allows the user to reuse these simple animations to build complex ones. Figure 5 shows the H-Animator interface in the joining phase: a combo-box on the upper-left corner allows one to select the animation category (e.g., teacher's gestures, American Sign Language). Categories are used to filter the list of available animations right below the combo-box.

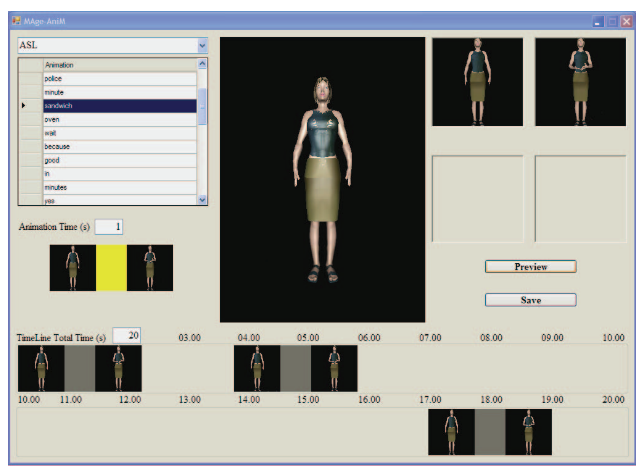

Figure 5. H-Animator interface for joining.

After selecting a simple animation from this list, H-Animator displays a simplified representation of the animation below the list. This representation is composed by three aligned elements: a miniature of the initial posture, a colored rectangle, a miniature of the final posture. The width of the colored rectangle is proportional to the duration of the animation. The user can click on this representation and then place a copy of it in the timeline at the bottom, so she can compose a complex animation sequence using the simple animations she needs. The user can see a preview of the partially modeled animation, played on the central window, whenever she wants. To produce the previews and the final animation, the chosen simple animations are retrieved from the database and the transitions between them are automatically generated by the system.

Transition generation is a problem for which the literature presents several solutions: a simple solution is to start and end each simple animation in the same neutral posture, so that no transition is needed. This solution may be good only if the embodied agent waits for some time in a given intermediate posture between each pair of subsequent simple animations. Otherwise, forcing an intermediate posture may lead to unrealistic animations: for example, in a sign language application, there are no pauses between signs in the same sentence, so this solution is unsuitable, as discussed in [38]. Complex solutions, such as using finite state automata [34] or Parallel Transition Networks [5], have been proposed in the literature, but we decided to choose a simpler semi-automatic linear interpolation technique that can be quickly understood by a novice user. Although this solution is simple, the generated transitions are realistic in most of the animations we built: for sign language only about 5 per cent of the animations were unrealistic and required to add more postures to get a realistic result. An example of unrealistic animation can be produced by concatenating a simple animation which ends with an arm behind the body and an animation which starts with the same arm in front of the body: the resulting linear transition compenetrates the agent torso. Anyway, it is easy for the user to predict if a pair of postures leads to an unrealistic transition: the result of our linear interpolation technique is always the shortest path from the first posture to the second one. The user can correct the problem by placing an intermediate posture between the involved postures.

After the joining process, the user can save the animation in a file which follows the X3D standard. Since no X3D players are currently available for mobile devices, we built the MobiX3D player described in the following subsection.

3.2 MobiX3D Player

Although the increasing performance of mobile computing devices such as Personal Digital Assistants (PDAs) or high-end mobile phones allows them to support more and more complex applications, displaying 3D animations in real-time on mobile devices is still difficult. Mobile devices are indeed characterized by some serious limitations affecting CPU, memory and graphics acceleration, which is often unavailable.

These limitations make it difficult to obtain frame rates high enough for smooth animation of complex embodied agents. If an animation is not smooth enough, the user can perceive it as unrealistic and the interface loses the typical advantages provided by embodied agents. Therefore, in our mobile applications, we use more simplified, but still realistic versions of the agents we use in desktop applications. To obtain smooth animations, we drastically reduce the size in triangles of the embodied agents (e.g. from about 20,000 to about 6,000) and eliminate the textures (the loss of detail is scarcely noticeable on the low-resolution mobile screen).

MobiX3D is a player based on the OpenGL Embedded System (OpenGL ES) [24] API. It displays X3D content on PDAs. Technical details about MobiX3D and how it uses OpenGL ES for rendering purposes are provided in [26]. MobiX3D Player can display both embodied agents and 3D environments where the agents are placed, provided that those environments are specified following the X3D standard.

We tested MobiX3D Player both on low-end (e.g. the Acer n10) and high-end PocketPCs (e.g. the Dell Axim X50V). The Dell X50V has a 624 MHz processor with 64 MB of main memory and an Intel 2700G graphics processor with 16 MB of video RAM. MobiX3D Player exploits use hardware graphics acceleration, provided by the Intel 2700G graphics processor, using the Intel implementation of the OpenGL ES API to display the animations of embodied agents smoothly (about 10 frames/second with a 6.000-triangles agent).

4. CASE STUDIES

4.1 Sign Language Agents on Mobile Devices

Sign languages, i.e. the visual communication languages used by deaf people, are an interesting application domain for embodied agents. For example, Karpouzis et al. [23] and Sagawa and Takeuchi [31] describe architectures and tools which use embodied agents to teach sign languages, while Zhao et al. [38] propose an automatic machine translation system from written English to American Sign Language. The ViSiCAST project [16] aims at improving the accessibility of public services, e.g. by using monitors which display embodied agents performing the sign language translation of what a Post Office counter clerk say. Yi et al. [37] proposed a tool to create and reuse animations for sign languages, using a specifically designed database.

For our project, we initially retrieved information on sign languages and deaf people needs from deaf associations and specialized forums, news and websites. It clearly emerges that deaf communities want to stress the linguistic importance of their own language, pointing out that different geographical areas use sign languages which differ in grammar and vocabulary and that each of these languages comprises thousands of signs. Moreover, some communities would like to use their sign language even in written communication (e.g., chat, e-mail): these communities are used to build sentences according to their sign language grammar, so writing in a spoken language is less natural.

Therefore, we extended MAge-AniM to allow the user to easily compose animations of sign language sentences: she can simply write sentences using the grammar of her sign language and the system retrieves from the database the simple animations corresponding to the words in the sentence. If there are no available animations for some words, the user can model them using the H-Animator tool. Anyway, even with a fast and easy-to-use tool, it would be an enormous task for a single user to produce all the animations for her sign language vocabulary, so MAge-AniM allows a community of users to share the simple animation database. Sharing may help deaf communities in populating their own vocabulary, significantly reducing the time needed.

We also added a fingerspelling function that shows a word letter-by-letter with finger movements when the word has no associated sign (e.g. family or geographical names).

Therefore, the sign language application can retrieve or build an animation for all the words and then produce the animation for the sentence. Users can view the animation in a preview window; they can change or correct the sentence and save an X3D file for it, ready to be sent, published and visualized.

However, displaying these sign language animations is a difficult task: if animations are not very precise, the user can misunderstand the meaning of the gestures. Moreover, when a word has to be fingerspelled, the animation is even more difficult to display due to the little size of fingers and the similarity of the movements associated to the letters. While on desktop devices the screen is wide enough to display the agent properly, displaying the animations on mobile devices in fullscreen mode is not enough. For this reason and limited performance considerations, no proposals on mobile devices are yet available.

In our project, we use the MobiX3D player to display sign language animations on mobile devices. To solve the fingerspelling issue, we change the viewpoint whenever a fingerspelled word animation is needed, by zooming on the signing hand, then we return to a half-length visualization. Figure 6 shows an embodied agent performing a sign language gesture on a PDA.

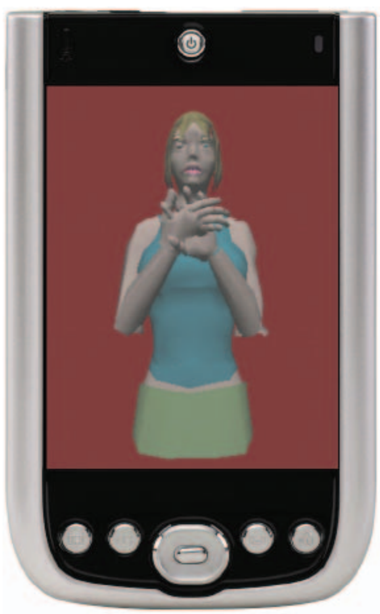

Figure 6. Mobile embodied agent performing sign language.

4.2 Mobile Fitness Training Agent

Fitness is another interesting application domain for embodied agents. In some indoor fitness applications [27][32], agents show how to correctly perform physical exercises and also provide advice to prevent injuries. To the best of our knowledge, there are no fitness applications that use embodied agents on mobile devices.

By using MAge-AniM, we developed a mobile animated agent which helps users to correctly perform fitness trail exercises. A fitness trail is a trail where the user has to alternate jogging and exercising. The user has to run along a path and has to stop when she arrives at an exercise station. In each exercise station, the user finds an exercise tool that she has to use to perform a specific fitness exercise. Exercises are usually explained by using illustrated plates in the stations. These plates are often difficult to understand and the exercise could be performed improperly. For this reason, when the user is near a fitness trail exercise, our PDA-based application displays the animated agent correctly performing the exercise.

Using MAge-AniM, we were able to quickly model exercise animations (using the H-Animator) tool and display them on PDAs (with the MobiX3D player). Figure 7 shows an example of fitness trail animation built and displayed using MAge-AniM.

For modeling fitness trail exercises, H-Animator offers the possibility to load a 3D world, specified by an X3D file, and to move the animated agent into the world. This functionality helps the user to model animations considering the agent's position and the various objects contained in a 3D world.

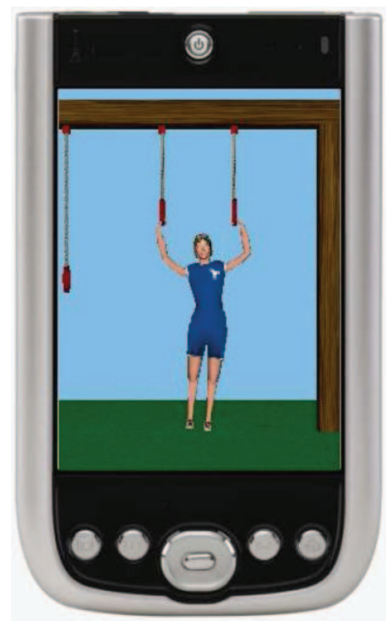

Figure 7. Mobile embodied agent doing a fitness exercise.

5. INFORMAL EVALUATION

We carried out a preliminary informal evaluation of MAge-AniM. To test the H-Animator tool easiness-of-use, we prepared a set of 5 photographs showing an actor assuming different postures, which require the rotation of seven joints. We involved 5 users with no previous experience in embodied agent animation (and animation in general). After briefly introducing the system to the users, we gave them the photographs and we asked them to model the postures with the system. The average time to model a posture was 3 minutes and 33 seconds. After they modeled all the 5 postures, we asked the users to build a simple animation starting from the pictures of the embodied agent postures. The average time to produce the animation was 1 minute and 5 seconds. These results are encouraging since it was the first time these inexperienced users tried the tool.

We tested the quality of the animations produced with MAge-AniM in the sign language as well as the fitness domain. For sign language animations, we involved three Italian Sign Language (LIS) experts. We quickly modeled some sign language sentences with H-Animator and then we showed them to the experts. They immediately and correctly recognized the meaning of the animated sign language sentences. The experts also suggested us some improvements to our sign language animations, such as adding facial expressions to represent the non-verbal sign language component. To test Mage-AniM fitness animations, we involved 10 users to perform some exercises in a local fitness trail. None of the 10 users had ever tried that fitness trail before. We asked 5 users to perform firstly an exercise after looking at the metal plate on the fitness trail station and then another exercise with the same difficulty after watching the animation on the PDA. The other 5 users firstly performed an exercise after watching our animation and then another exercise after looking at the metal plate. All 10 users correctly performed their exercises after watching the animations and only 2 of them performed their exercise correctly after looking at the metal plate.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the MAge-AniM system for the visual modeling of embodied agent animations and their reply on mobile devices. Since MAge-AniM is intended for novice users, we based it on a familiar metaphor and we provide animation previews and instant feedback about postures.

The current release of MAge-AniM is freely available at [18] and we are currently working on a new release of the system which will offer new features. In particular, to further simplify the posing phase, we are working at introducing inverse kinematics combined with a visual manipulation system. To add more realism to the agent animations on the mobile player, we will test if seamless shape deformation algorithms (i.e. parametric algorithms which allow one to use a single deformable mesh for the agent instead of using several separated meshes) can be adapted for mobile devices without significant decrease of performance.

Finally, we will formally evaluate the whole system. On one side, we will test H-Animator asking both hearing and deaf users to model simple and complex animations using sample databases, measuring the time they took to perform the task and comparing these results with those obtained using other methods. On another side, we will test the quality of the animations on PDAs with several users. In particular, we will follow two directions: (i) for sign language, we will verify if the animations are easily understood by the users, asking them to translate the animations in written sentences; (ii) more generally, we will evaluate the users' perceived realism of animations through questionnaires.

7. ACKNOWLEDGEMENTS

Our research has been partially supported by the Italian Ministry of Education, University and Research (MIUR) under the PRIN 2005 project "Adaptive, Context-aware, Multimedia Guides on Mobile Devices".

8. REFERENCES

[1] B. Adelson. Evocative Agents and Multi-Media Interface Design. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 351–356. ACM Press, New York, NY, USA, 1992.

[2] Alias-Wavefront. Maya. http://www.aliaswavefront.com/en/products/maya/, 2005.

[3] E. André, T. Rist, and J. Müller. Guiding the User through Dynamically Generated Hypermedia Presentations with a Life-Like Character. In *IUI '98: Proceedings of the 3rd international conference on Intelligent user interfaces*, pages 21–28. ACM Press, New York, NY, USA, 1998.

[4] Autodesk Inc.. 3ds Max. http://www.discreet.com/products/3dsmax/, 2005.

[5] N. I. Badler, M. S. Palmer, and R. Bindiganavale. Animation Control for Real-time Virtual Humans. *Commun. ACM*, 42(8):64–73, 1999.

[6] J. Baus, K. Cheverst, and C. Kray. A Survey of Map-based Mobile Guides. In L. Meng, A. Zipf, T. Reichenbacher (eds): *Map-based mobile services - Theories, Methods, and Implementations*, pages 197–216. Springer-Verlag, Berlin, Germany, 2005.

[7] T. Bickmore. Towards the Design of Multimodal Interfaces for Handheld Conversational Characters. In *CHI '02: Extended abstracts on Human factors in computing systems*, pages 788–789. ACM Press, New York, NY, USA, 2002.

[8] T. W. Bickmore and R. W. Picard. Establishing and Maintaining Long-term Human-Computer Relationships. *ACM Trans. Comput.-Hum. Interact.*, 12(2):293–327, 2005.

[9] Bitmanagement Software GmbH. BS Contact VRML/X3D. http://www.bitmanagement.de, 2005.

[10] F. Buttussi, L. Chittaro, and D. Nadalutti. H-Animator: A Visual Tool for Modeling, Reuse and Sharing of X3D Humanoid Animations. In *Web3D '06: Proceedings of the 11th international conference on 3D Web technology*. ACM Press, New York, NY, USA, 2006.

[11] M. P. Carretero, D. Oyarzun, A. Ortiz, I. Aizpurua, and J. Posada. Virtual Characters Facial and Body Animation through the Edition and Interpretation of Mark-up Languages. *Computers and Graphics*, 29:189–194, 2005.

[12] J. Cassell, T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, H. Vilhjálmsson, and H. Yan. Embodiment in Conversational Interfaces: Rea. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 520–527. ACM Press, New York, NY, USA, 1999.

[13] J. Cassell, T. Bickmore, L. Campbell, H. Vilhjálmsson, and H. Yan. Human Conversation as a System Framework: Designing Embodied Conversational Agents. In J. Cassell, J. Sullivan, E. Churchill, S. Prevost (eds): *Embodied conversational agents*, pages 29–63. MIT Press, Cambridge, MA, USA, 2000.

[14] M. Conway, S. Audia, T. Burnette, D. Cosgrove, and K. Christiansen. Alice: Lessons Learned from Building a 3D System for Novices. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 486–493. ACM Press, New York, NY, USA, 2000.

[15] J. Davis, M. Agrawala, E. Chuang, Z. Popovic, and D. Salesin. A Sketching Interface for Articulated Figure Animation. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer*

animation, pages 320–328. ACM / Eurographics, Aire-la-Ville, Switzerland, 2003.

[16] R. Elliott, J. R. W. Glauert, J. R. Kennaway, and I. Marshall. The Development of Language Processing Support for the ViSiCAST Project. In *Assets '00: Proceedings of the 4th international ACM conference on Assistive technologies*, pages 101–108. ACM Press, New York, NY, USA, 2000.

[17] M. Gutiérrez, F. Vexo, and D. Thalmann. The Mobile Animator: Interactive Character Animation in Collaborative Virtual Environments. In *VR '04: Proceedings of the IEEE Virtual Reality 2004*, pages 125–132. IEEE Computer Society, Washington, DC, USA, 2004.

[18] HCI Lab - University of Udine, MAge-AniM website, http://hcilab.uniud.it/mage-anim/.

[19] A. Huang, Z. Huang, B. Prabhakaran, and J. C. R. Ruiz. Interactive Visual Method for Motion and Model Reuse. In *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 29–36. ACM Press, New York, NY, USA, 2003.

[20] Humanoid Animation Working Group. ISO/IEC FDIS 19774 Humanoid Animation (H-Anim). http://h-anim.org, 2001.

[21] M. W. Kadous and C. Sammut. InCA: A Mobile Conversational Agent. In *PRICAI 2004: 8th Pacific Rim International Conference on Artificial Intelligence*. *Lecture Notes in Computer Science*, volume 3157, pages 644–653. Springer-Verlag, Berlin, Germany, 2004.

[22] K. Kamyab, F. Guerin, P. Goulev, and E. Mamdani. Designing Agents for a Virtual Marketplace. *AISB Journal*, 1(1):61–86, 2001.

[23] K. Karpouzis, G. Caridakis, S. E. Fotinea, and E. Efthimiou. Educational Resources and Implementation of a Greek Sign Language Synthesis Architecture. *Computers & Education, Special Issue in Web3D Technologies in Learning, Education and Training*, In press.

[24] Khronos Group. OpenGL ES. http://www.khronos.org/opengles/, 2003.

[25] J. C. Lester, S. A. Converse, S. E. Kahler, S. T. Barlow, B. A. Stone, and R. S. Bhogal. The Persona Effect: Affective Impact of Animated Pedagogical Agents. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 359–366. ACM Press, New York, NY, USA, 1997.

[26] D. Nadalutti, L. Chittaro, and F. Buttussi. Rendering of X3D Content on Mobile Devices with OpenGL ES. In *Web3D '06: Proceedings of the 11th international conference on 3D Web technology*. ACM Press, New York, NY, USA, 2006.

[27] responDESIGN, Inc. Yourself!fitness. http://www.yourselffitness.com/.

[28] T. Rist, E. André, and S. Baldes. A Flexible Platform for Building Applications with Life-Like Characters. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, pages 158–165. ACM Press, New York, NY, USA, 2003.

[29] T. Rist, E. André, and J. Müller. Adding Animated Presentation Agents to the Interface. In *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 79–86. ACM Press, New York, NY, USA, 1997.

[30] Z. Ruttkay and C. Pelachaud. From Brows to Trust: Evaluating Embodied Conversational Agents (Human-Computer Interaction Series). Springer-Verlag, Berlin, Germany, 2004.

[31] H. Sagawa and M. Takeuchi. A Teaching System of Japanese Sign Language Using Sign Language Recognition and Generation. In *MULTIMEDIA '02: Proceedings of the 10th ACM international conference on Multimedia*, pages 137–145. ACM Press, New York, NY, USA, 2002.

[32] Sony Computer Entertainment Europe and Nike Motion Works. Eyetoy: Kinetic. http://www.eyetoykinetic.com/.

[33] S. C. L. Terra and R. A. Metoyer. Performance Timing for Keyframe Animation. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 253–258. ACM Press, New York, NY, USA, 2004.

[34] L. van Zijl and L. Raitt. Implementation Experience with Collision Avoidance in Signing Avatars. In *AFRIGRAPH '04: Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualization and interaction in Africa*, pages 55–59. ACM Press, New York, NY, USA, 2004.

[35] Web3D Consortium. ISO/IEC 19775:2004 Extensible 3D (X3D). http://www.web3d.org/x3d/specifications/, 2004.

[36] S. Wilcox, J. Scheibman, D. Wood, D. Cokely, and W. C. Stokoe. Multimedia dictionary of American Sign Language. In *Assets '94: Proceedings of the 1st annual ACM conference on Assistive technologies*, pages 9–16. ACM Press, New York, NY, USA, 1994.

[37] B. Yi, J. Frederick C. Harris, and S. M. Dascalu. From Creating Virtual Gestures to "Writing" in Sign Languages. In *CHI '05: Extended abstracts on Human factors in computing systems*, pages 1885–1888. ACM Press, New York, NY, USA, 2005.

[38] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. I. Badler, and M. Palmer. A Machine Translation System from English to American Sign Language. In *AMTA '00: Proceedings of the 4th Conference of the Association for Machine Translation in the Americas on Envisioning Machine Translation in the Information Future*, pages 54–67. Springer-Verlag, Berlin, Germany, 2000.