

Animating 2D Digital Puppets with Limited Autonomy

Erin Shaw, Catherine LaBore, and Yuan-Chun Chiu

Center for Advanced Research in Technology for Education
University of Southern California, Information Sciences Institute,
4676 Admiralty Way, Marina del Rey, 90292, USA
{shaw, labore, yuanchuc}@isi.edu
<http://www.isi.edu/isd/carte/index.html>

Abstract. In this paper we describe the graphical realization and authoring of a semi-autonomous animated persona, called a Digital Puppet. Digital Puppets augment on-line educational materials with assistance, commentary and summaries, and have been used in both the tutoring and presentation domains. Limited animation, 2D "cartoon" characters are useful for some pedagogical applications, particularly for young learners, but production demands frequently exceed limited research budgets. To make the production of 2D personas feasible, we present a build-once, use-forever production path that allows us to quickly create new behaviors for characters. Animations are generated by selecting behaviors based on textual content and presentation and tutorial goals, allowing the puppet to act with limited autonomy within specific domains.

1 Introduction

Animated characters are becoming increasingly popular for use in conversational interfaces, and as presentation and pedagogical agents. Learning research by Mayer and Moreno has shown that voice over narration and personalized commentary are particularly effective multimedia presentation methods [16][17]. Such presentations help learners to make connections among representations presented in visual and auditory modalities. This helps learners to organize information into a coherent knowledge structure, resulting in deeper learning. Furthermore, animated agents such as Herman the Bug can produce a positive affective response in the viewer, sometimes referred to as the *persona effect*, which states that the presence of a life-like character - even one that is not expressive - can have a positive effect on the student's perception of a learning experience [12].

The work builds upon our work in the area of animated pedagogical agents, or "guidebots" [11][7][20]. Guidebots interact with learners via a combination of speech and gestures, making it possible to more accurately model the kinds of dialogs and interactions that occur during apprenticeship learning and one-on-one tutoring. They can demonstrate visually how to perform tasks, and provide ongoing commentary about the material. They personify the interface, building upon people's natural tendency to interact socially with computers [19], and can express both thoughts and emotions, in order to portray enthusiasm and empathy. Like real expert tutors that

attend to both motivational and cognitive factors, animated characters have the potential to increase learner curiosity and interest, and to offer help and reassurance when they encounter difficulties [19].

In this paper we describe the graphical realization and authoring of a semi-autonomous animated persona, called a Digital Puppet. Digital Puppets augment on-line educational materials with assistance, commentary and summaries, and have been used in both the tutoring and presentation domains. Limited animation, 2D "cartoon" characters are useful for some pedagogical applications, particularly for young learners, but production demands frequently exceed limited research budgets. To make the production of 2D personas feasible, we present a build-once, use-forever production path that allows us to quickly create new behaviors for characters. Animations are generated by selecting behaviors based on textual content and presentation and tutorial goals, allowing the puppet to act with limited autonomy within specific domains.

2 Related work

Work in the area of animated characters is broad and deep. Scriptable online characters such as those from Oddcast, Haptek, Microsoft have become increasingly common in the commercial domain where they act as social interfaces or guides [3][18][8]. Because each utterance and gesture must be specified, and because the associated graphics must be kept simple, these interface aids are of limited general use. More sophisticated commercial applications such as those by Conversive and EDrama apply templates and rules to alleviate repetitiveness and provide a degree of autonomy [5][6]. Whereas commercial applications of animated characters assume simple scripting and graphics, research applications explore multimedia presentation and discourse planning, and animating complex behaviors [2][25][12].

The work presented here fits right in the middle. Digital puppets are scripted and sequenced, much like in AutoTutor 2 [25][26], though the puppet-scripting model allows for greater flexibility, supporting parallel actions, behavior choices and behavior probabilities. The main difference, however, is the use of presentation goals to create semi-autonomous gestures. In this respect we are similar to Andre and Rist's PPP Persona, a multipurpose presentation agent. [1][2] However, puppets are not intelligent or autonomous agents because they don't plan. Though they are less powerful within any particular domain, it is easy to script puppet for new domains.

The work presented here is built upon our experiences creating animated characters for pedagogical and clinical purposes [14][15][20][24], and compatible with work on automated behavior expression, especially Cassell, Vilhjalmsen, and Bickmore's BEAT [4], which would fit nicely into the puppet system. For the moment, we have created a very course-grained version of this system.

3 3D Character Creation and Animation

3.1 Graphical Realization

The prototypes for the character animation system were Adele [20], the tutor for our case-based clinical practice training simulation, and Carmen and Gina, characters in the interactive drama, Carmen's Bright Ideas [14], that was built to teach the Bright Ideas problem-solving method to mothers of pediatric cancer patients.

Adele's persona is animated using 2D sequences of frames, always returning to a default position after performing a gesture, and employing classic limited animation techniques. Adele was built using hand-drawn keys and in-betweens, rendered directly by an artist using a standard bitmap paint program. A library of this artwork is loaded into the simulation. The Adele agent then has a persona, which can perform a limited number of very generic gestures.

Carmen's Bright Ideas uses traditional hand-drawn character animation, realized as 2D computer graphics (.swf) using Macromedia's Flash. The agent-controlled characters, Carmen and Gina, use composite Flash libraries for various gestures, head and eye positions, and facial expressions. The shockwave files are imported into Director, and are requested by the agent using Active-X controls.

The 2D behavior-library approach is useful in a number of contexts. Many agent domains are limited in their requirements for visual behavior, relying on synthetic speech to provide necessary depth and breadth. Applications that use a 2D interface look flat in the context of a 3D character, and real-time rendering to line-and-fill art is technically challenging. The limited-animation, 2D "cartoon" character as the basis for a persona seems useful and viable for pedagogical applications, particularly for young learners.

Such libraries are most useful when we can load and unload components as needed. In the case of hand-drawn art, if the agent's tutorial domain changes, or requires additional behaviors, the same artist must be re-engaged, and spend many more hours in the labor-intensive task of hand drawing the new frames.

Research budgets and circumstances make this approach less than ideal. To make our 2D personas feasible, we opted for a "build-once, use-forever" production path. The character for the first Digital Puppet system, Skip, was built using a commercial 3D animation program. Any competent character animator can create new behaviors for Skip. Rendered output will always match the original art, so libraries of behavior can be built up over time. In addition, a procedural system such as Alias-Wavefront's Maya allows for the generalization of behaviors across characters.

In addition, we can render the character in layers, allowing us to achieve independent control, in the animation engine, to maximize behavioral capability. Eyes, brows, mouth and body can each be rendered separately so that the character can while raise its brows while speaking, speak while gesturing, shift its gaze and weight when attending to user actions, etc.

Behaviors can thus be created to meet the evolving needs of the agent. We began with a basic library that makes the character come alive: weight and gaze shifts, sac-

cares, blinks and head movements. In the case of the Presentation agent, Skip, behaviors specific to expository speaking and the use of visual aids are added on top of these. Future additions to Skip might add tutorial gestures related to orienting the user to web-lesson annotation, how to use the application interface, and other related tasks.

For additional flexibility, we used scalable vector graphics (svg) as our target display format. This allows us to scale the renders to the optimal size for the screen layout and employ a variety of “cameras,” from full-body shots for gestures to close-ups for conversational behaviors. Animation files are created, automatically queued for render at base body orientations (toward the user, toward the screen) using MEL scripts. Render output is to Macromedia’s 2D vector-animation application Flash (.swf), which provides indexing and editing capabilities. The images are exported to an Adobe image format (.ai) and then converted to scalable vector graphics (.svg.)

3.2 Creating Behaviors

Beyond the basic behavior library, the animators must create behaviors related to a particular domain that will be used by the character, such as presentation. Video footage of human presenters was collected and studied. Deictic gestures combine with beats in presentations as the most common gestures in presentation contexts. Deictics are easy to define procedurally and generically useful. We have also collected video footage from human tutors working with learners using computer applications. This reference footage permits the animators to anticipate what kinds of moves will be indispensable.

3.3 Behavior Descriptions

Libraries of animation images are ultimately loaded into the puppet animation engine. XML descriptions for various gestures are defined as in Table 1, using a combination of sequenced and procedural frames. The animation uses the definitions to trigger the desired behavior. Background animations are automatically called when the agent is not requesting specific behavior. Sending a text string for speech synthesis automatically animates lip sync at any of several different head positions.

Table 1. Examples of facial animations and gestures.

Agent: “body-turnR” = turn body from user orientation to screen orientation
Agent: “gaze10” = set gaze to 10 o’clock position
Idle behavior: blinks, slight head and body movements
Agent: “doubt” = run facial animation sequence “doubt”
Agent: “turn-headL” = turn face toward user
Agent: “head009tl” = tilt head left
Agent: “turn-headR” turn face toward screen
Agent: “point9” point to 9 o’clock
Idle behavior: blinks, slight head and body movements

4 Animation Authoring

There are two ways in which we map contextual hints to behaviors. In the two domains we have discussed, there are naturally occurring high-concept tasks. In the presentation domain there is there are the components of the presentation task (e.g., introduction, conclusion) and in the tutoring domain there is the flow of the application task (e.g., select a factory, run forecasting.) The animation markup for scripting a puppet at this level is shown in Table 2.

Table 2. Mapping high-level contextual information to facial behaviors and gestures.

<pre> <context name="PREINTRO"> <s type="camera" name="closer"/> <s type="gspeak" text="Come { %talkGesture:default% } in and sit down."/> <s type="wait" for="second" value="5"/> <s type="gspeak" text="We { %turnhead:044% } are still waiting for people to arrive."/> <s type="choices"> <c type="pose" gesture="talkGesture" subtype="default"/> <c type="nothing"/> <c type="group"> <g type="turnhead" to="044" wait="true"/> <g type="turnhead" to="000" wait="true"/> <g type="turnhead" to="044" wait="true"/> </c> </s> <s type="wait" for="second" value="5"/> <s type="jump" context="PREINTRO" /> </context> </pre>	<pre> <context name="INTRODUCTION"> <s type="turnhead" to="044" wait="true"/> <s type="wait" for="second" value="1"/> <s type="turnhead" to="000" wait="true"/> <s type="turnbody" to="000" wait="true"/> <insert_point> </insert_point> </context> </pre>
--	--

The *preintro* is the initial context, and in this state the camera will zoom in and the puppet will occasionally look around and speak. When the viewer presses the introduction button, the puppet goes into the *introduction* state, which mandates that the puppet first glance at the button, and then orient itself with its body facing forward before it speaks. There are *idle* and *post-conclusion* states as well. The type “choices” will choose one of the gestures, or one group, while the type “gspeak” will specify a gesture to perform while speaking.

The commands are executed asynchronously and in parallel. For example, the “speak” command might take several seconds to complete but won’t block the scripting engine from executing the next command. The “wait” command is used to force the engine to wait for the “speak” command to finish before executing the next command or to wait for a given period of time. Speaking itself is the one exception, since the puppet can only speak sentences sequentially.

The more interesting mapping of behaviors occurs at lower levels, where gestures are mapped to contexts and goals (e.g., “elaborate” in the presentation domain, and mitigate face threat in the “tutoring” domain), sentence fragments, words, and patterns, such as exclamations and numbers. This intention-based mapping is similar to the approach of Andre and Rist [2], who use presentation goals to select their anima-

tion schema, and to Cassell et al. [4]. The gesture is executed randomly after the sentence starts, with the probability that it will occur being specified.

Table 2. Mapping low-level contextual information to facial behaviors and gestures.

<pre> <word> <entry word = "this" gesture = "lifthandpoint" subtype = "default" probability = "100" /> <entry word = "but! but," gesture = "shiftweightright" subtype = "default" probability = "50" /> </word> <sentence> <entry fragment = "in fact" gesture = "talkgesture2" subtype = "default" probability = "100" /> </sentence> </pre>	<pre> <context> <entry context = "INTRODUCTION" gesture = "talkgesture1" subtype = "default" probability = "50" /> </context> <goal> <entry goal = "claim" gesture = "lookaround" subtype = "default" probability = "50" /> </goal> </pre>
---	--

5 Puppet Applications

5.1 Puppets for developing oral presentation skills

Digital puppets were created as part of an application to teach fourth and fifth grade students how to construct oral presentations. The goal of the project was to improve science learning through the use of *digital puppets* in peer teaching and collaborative learning settings. Because of the user base, the user-friendly WYSIWYG puppet-authoring tool, shown in Figure 1 on the left, was developed in parallel with the animation engine, instead of as an afterthought. The tool enables users to annotate Web pages and use the annotations to generate commentary and animation.

Our approach to authoring the puppets [24] was based on Rhetorical Structure Theory (RST) [13] but general enough to accommodate any specified goal. The presentation domain was constrained enough to allow inferences to be made about the intent of the author, with respect to the authored commentary. For example, if the puppet was authored to introduce a subject in order to motivate it, we mapped the context of the input, i.e. "introduce", and its goal, i.e. "motivate", to a set of appropriate animations, e.g., face recipient, perform inviting gesture, amplify voice, and pause.

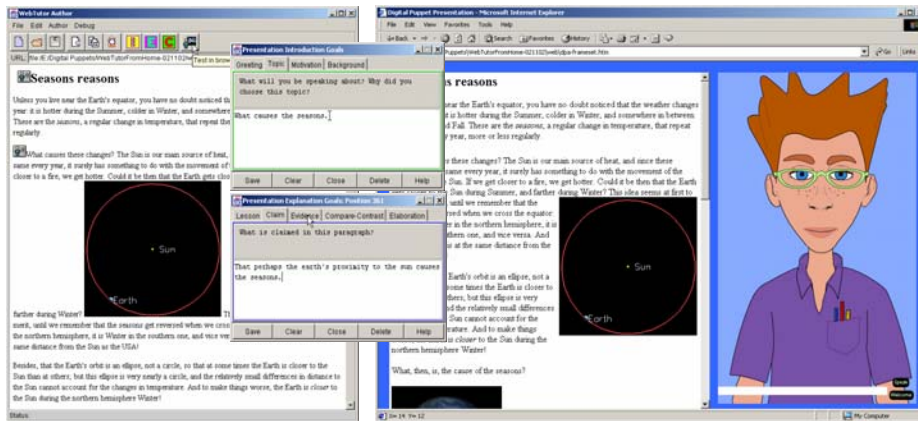


Figure 1. The Digital Puppet authoring interface (left) and Skip-puppet-enhanced Web page

Following the flow diagram in Figure 2, the Web page to be annotated is loaded into the authoring tool. Working in small groups, students use the tool to write an introduction, a conclusion and several paragraph-linked explanations that, when played together, result in an oral presentation about the subject matter. For example, clicking on a paragraph of text brings up an authoring window like the one shown in Figure 3 that displays the particular concepts that the students must address in their explanations, as well as more detailed instructions and prompts. These goals (e.g., lesson, claim, evidence) and instructions can be tailored for different contexts (i.e., introduction, explanation, and conclusion.)

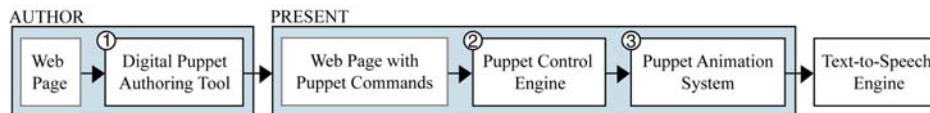


Figure 2. Flow diagram of the puppet authoring and animation system.

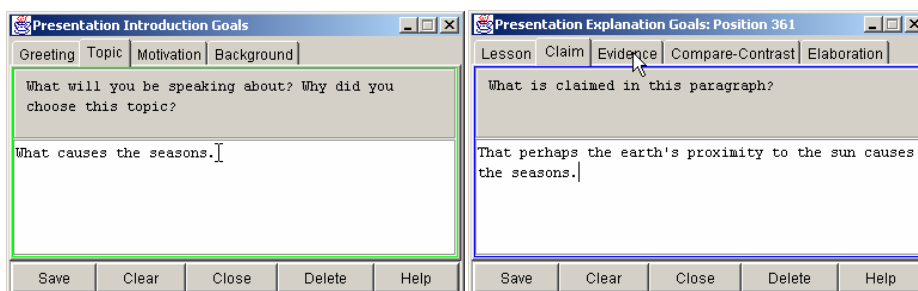


Figure 3. Authoring windows for an introduction and paragraph-level explanation.

The user publishes the resulting puppet-enhanced Web page by clicking on a button. The published page contains the presentation text (authored input) and contextual

hints (presentation goals), HTML-based puppet commands and interactive buttons, and the JavaScript-based puppet control engine. The Web page and Java-based puppet applet are automatically displayed in adjacent browser frames, as shown in Figures 1 (right) and 2. The user activates the presentation by pressing an interactive button.

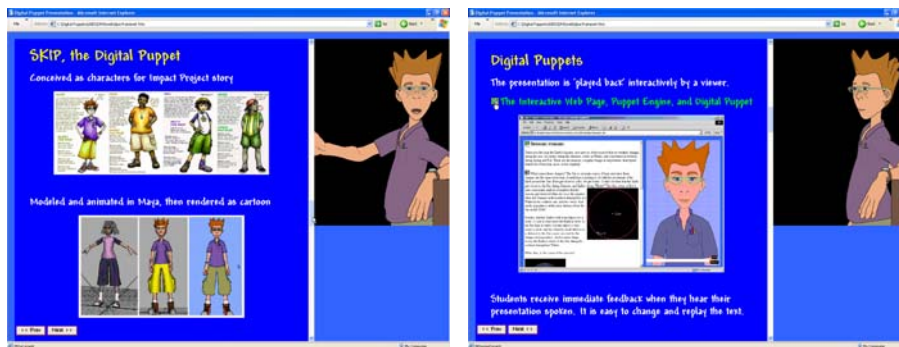


Figure 4. The VFTS application (left), the Chat applet (right, top) with *Widget*, and the tutorial browser window (right, bottom).

5.2 Wizard of oz study for Socially Intelligent Agents project

Digital puppets are also being used in a wizard of oz study as part of a research project on socially intelligent agents (SIAGents) in educational domains, to assess the appropriateness of the output of a dialogue generator. In this study, a domain expert watches a student perform a task via the interface of the Virtual Factory Teaching system, an educational factory modeling application and simulation to teach undergraduate engineering students production and inventory management. If the expert chooses to initiate a speech act, in order to comment or give advice, he selects an appropriate topic and verbosity level from the wizard of oz interface and an utterance is generated and sent to the user.

As in the presentation system, the generated text is sent verbatim to the puppet. To generate the animation requires contextual information that allows us to map behaviors to textual input. The SIAGents system is based on the Politeness Theory of Brown and Levinson [], and uses that model to influence the student's motivational state. The system has a rich user model, which influences the generated dialogue. For example, if the system determines that a user is frustrated, it can initiate dialogue to mitigate a face threat. The politeness factor was therefore used as a context for mapping behaviors. The system also has an extensive plan, and can tell if user is deep in a task, or only beginning or concluding one. The plan provided another context within which to script behaviors. The model is extensible, so that one can use as much or as little context as required to influence a puppet's animation behaviors.

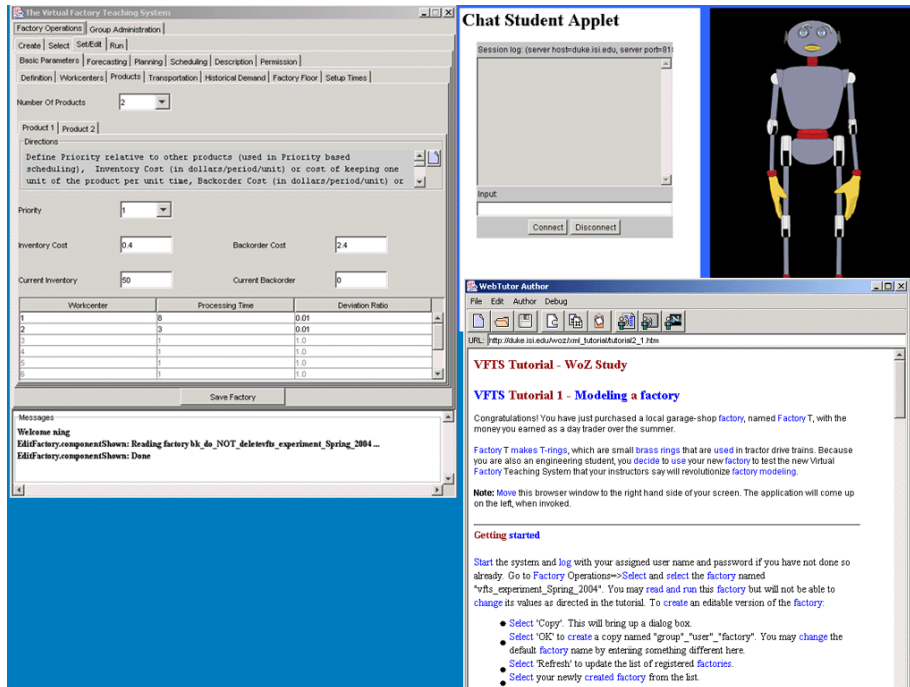


Figure 5. The VFTS application (left), the Chat applet (right, top) with puppet *Widget*, and the tutorial browser window (right, bottom).

6 Conclusion

We have presented a new production and authoring pipeline for animating a semi-autonomous character, the digital puppet, which can act with limited autonomy within constrained domains. To allow us to quickly create new behaviors for characters and make the production of 2D personas feasible, we developed a build-once, use-forever production path. We have shown how a digital puppets are used in both tutoring and presentation contexts. As computing power increases and on-line animated characters learn to behave more appropriately, that is, as believability improves, and conversational interfaces will become an interface standard. We will see 2D and 3D, realistic and stylistic, and non-human puppets. Because reasoning in complex domains will remain a hard problem for much time to come, we expect semi-autonomous agents such as the digital puppet to bridge the gap between narrators and intelligent tutors.

References

1. André, E.: The Generation of Multimedia Presentations.

2. André, E., Rist, T., and Müller, J.: Integrating reactive and scripted behaviors in life-like presentation agents. In K.P. Sycara and M. Wooldridge (Eds.), *Proc. of the Second Int'l Conf. on Autonomous Agents*, pp. 261-268, ACM Press, New York. (1998)
3. Ball, Ling: Lifelike computer characters: The persona project at Microsoft (1996)
4. Cassell, J., Vilhjalmsson, H., and Bickmore, T.: BEAT: the Behavior Expression Animation Toolkit. SIGGRAPH Computer Graphics Proceedings, AMC Press (2001)
5. Conversive: InstantAgents, <http://www.conversive.com>
6. EDrama Learning: White paper, Natural language in simulated conversation. (2002)
7. Ganeshan, R., W. L. Johnson, E. Shaw, and B.P. Wood, Tutoring Diagnostic Problem Solving. In Proceedings of the Fifth Int'l Conf. on Int. Tutoring Systems, 2000
8. Haptik: Automated Personalities, <http://www.haptik.com>
9. Johnson, W.L. (2002). Using agent technology to improve the quality of Web-based education. In N. Zhong & J. Liu (Eds.). *Web Intelligence*. Berlin: Springer.
10. Johnson, W.L. and Shaw, E. (2001). K12 Agents: USC/ISI grant # 12-1540-0002.
11. Johnson, W.L., Rickel, J.W., and Lester, J.C.: Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education* 11, 47-78. (2000)
12. Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A., and Bhogal, R. S. (1997a). The persona effect: Affective impact of animated pedagogical agents. In *Proceedings of CHI '97*, 359-366. (1997)
13. Mann W. C. and Thompson S. A.: Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, ISI, Univ. of Southern Cal. (1987)
14. Marsella, S., W. L. Johnson, and C. M. LaBore, Interactive Pedagogical Drama. In Proceedings of the 4th Int'l Conference on Autonomous Agents, Agents 2000
15. Marsella, S., W. L. Johnson, and Catherine M. LaBore, Interactive Pedagogical Drama for Health Interventions. In 11th Int'l Conf. on AI in Education. (2003)
16. Mayer, R.E.: *Multimedia learning*. New York: Cambridge University Press. (2001)
17. Moreno, R., Mayer, R.E., & Lester, J. C.: Life-like pedagogical agents in constructivist environments: cognitive consequences of their interaction. In Bourdeau, J. & Heller, R. (Eds.), In Proceedings of ED-MEDIA 2000, 741-746, AACE. (2000).
18. Oddcast: SitePal, <http://vhost.oddcast.com> (1996)
19. Reeves, B. and Nass, C.: *The media equation*. Cambridge University Press. (1996)
20. Shaw, E.; Ganeshan, R.; Johnson, W. L.; and Millar, D.: Building a case for agent-assisted learning as a catalyst for curriculum reform in medical education. In Proceedings of the Ninth Int'l Conf. on AI in Education. IOS Press. (1999)
21. Shaw, E.; Johnson, W. L.; and Ganeshan, R.: Pedagogical agents on the Web. In Proceedings of the Third International Conference on Autonomous Agents. (1999)
22. Stone, Lester: Dynamically sequencing an animated pedagogical agent. (1996)
23. Rist, T., Andre, E., and Muller, J.: Adding Animated Presentation Agents to the Interface. In Proceedings of the 1997 International Conference On Intelligent User Interfaces, pages 79—86, Orlando, Florida. (1997)
24. Rizzo, P., Shaw, E., & Johnson, W.L.: An Agent that Helps Children to Author Rhetorically-Structured Digital Puppet Presentations. In Proceedings of ITS. (2002)
25. Drumwright, E. Putting the Sizzle in AutoTutor2. Report.
26. Person, N.K., Craig, S., Price, P., Hu, X., Gholson, B., Graesser, A.C., and the Tutoring Research Group: Incorporating Human-like Conversational Behaviors into AutoTutor. IUI (2003)
27. Lepper, M.R. & Henderlong, J. (2000). Turning “play” into “work” and “work” into “play”: 25 years of research in intrinsic versus extrinsic motivation. In C. Sansone & J.M. Harackiewicz (Eds.), *Intrinsic and Extrinsic Motivation: The Search for Optimal Motivation and Performance*, pp. 257-307. San Diego: Academic Press.