

Animating 2D Digital Puppets with Limited Autonomy

Erin Shaw, Catherine LaBore, Yuan-Chun Chiu, and W. Lewis Johnson

Center for Advanced Research in Technology for Education
4676 Admiralty Way, Marina del Rey, 90292, USA
{shaw, labore, joechiu, johnson}@isi.edu
<http://www.isi.edu/isd/carte/index.html>

Abstract. Digital puppets are animated personas that augment online educational materials with commentary and summaries. Their animations are generated dynamically based on user-authored text, contextual hints, and domain goals, allowing the puppet to act with limited autonomy within specific domains. In this paper we describe the graphical realization and authoring of a 2D digital puppet. We present a *build-once, use-forever* production path that allows us to quickly create new character behaviors, and makes the production of 2D personas feasible. We describe two digital puppet applications and explain how the animation capability is supported across domains.

1 Introduction

Animated characters are becoming increasingly popular for use in conversational interfaces, and as presentation and pedagogical agents. In this paper we describe the graphical realization and authoring of a animated character, or Digital Puppet, which can act with limited autonomy in its particular domain. Digital Puppets augment online educational materials with commentary and summaries, and have been used in both the tutoring and the presentation domain. The work presented here is builds upon our experiences creating pedagogical animated characters, or *guidebots* [19][20][24][28][12][15]. Like guidebots, digital puppets 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 personify the interface, building upon people’s natural tendency to interact socially with computers [23], 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 [16].

The digital puppet system, and in particular, its animation and authoring, is motivated by two drawbacks to developing agents in a research environment: limited budgets and the prototype development cycle. First, we have found limited-animation 2D characters particularly useful for some pedagogical applications; cartoon 2D for younger children and realistic 2D for adult storytelling. However, production demands for animated characters frequently exceed limited research

budgets. This is especially true for complex characters, like the puppets, that are designed to speak and gesture in parallel. To make the production of 2D personas feasible, we have developed a *build-once, use-forever* production path that allows us to quickly create new behaviors for a given character [set]. Second, we have found that the prototype iteration cycle for agents precludes a broad focus: Only after completion of the prototype is thought given to re-purposing the agent. XML has mitigated the problem, but hasn't removed the challenge of layperson authoring. Digital puppets were designed to have an integrated user-friendly authoring environment. We wished to separate the content authoring and the context authoring, and did not want to distract users with low-level animation details. Puppets speak exactly the lines they are authored to speak but their gestures and facial animation are generated automatically from contextual hints; thus, the puppets act with limited autonomy.

2 Related Work

With respect to authoring, animated characters fall into three categories, sequentially scripted characters, limited autonomous characters, and autonomous planning agents. 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 [4][13][21]. 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 [9][11]. Whereas commercial applications of animated characters assume simple scripting and graphics, research applications explore multimedia presentation and discourse planning, and the animation of complex behaviors[3][17][10].

Digital puppets are limited autonomous characters. They are scripted and sequenced, much like in AutoTutor 2 [10][22], 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 autonomous gestures. In this respect our puppet is similar to Andre and Rist's PPP Persona, a multipurpose presentation agent [2][3]. However, puppets do not plan, and thus are not intelligent or autonomous agents. Though they are less powerful within any particular domain, it is fairly efficient to transfer puppets to new domains.

The work presented here is compatible with work on automated behavior expression, especially the BodyChat and BEAT systems [6][7]. In BodyChat, gaze behaviors, for example, are fully automated for turn-taking in multi-participant settings. Though the Digital Puppets' behavioral domain is not social communication per se, some of its behaviors are similarly automated, and could be further developed using BEAT. Gestures signifying attentiveness to user actions, deictic gestures indicating on-screen objects such as lesson pages or simulation interfaces, and idle behaviors such as weight and gaze shifts, blinks and head

movements are automated. These automatic behaviors may be overridden by authored behaviors, e.g., those which are attached to content-free initiators and responses, in the case of the tutorial agent, or gestures chosen to accompany presentation material, in the case of the presentation agent.

3 2D Character Creation and Animation

3.1 Graphical Realization

The prototypes for the character animation system were Adele [25], the tutor for our case-based clinical practice training simulation, and Carmen and Gina, characters in the interactive drama, Carmen's Bright Ideas [19], which teaches problem-solving skills to mothers of pediatric cancer patients. Adele uses classic limited animation techniques - 2D sequences of frames that always return to a default position. Adele's persona was rendered directly by an artist using a commercial paint program. Carmen's Bright Ideas uses traditional hand drawn character animation, realized as 2D computer graphics using Macromedia's Flash. The agent-controlled characters, Carmen and Gina, use composite Flash libraries for gestures, head and eye positions, and facial expressions.

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 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 Alias Wavefront's Maya, a commercial 3D animation program. Any competent character animator can create new behaviors for Skip, and 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 Maya allows for the generalization of behaviors across characters.

To maximize behavioral capability, we render and animate the character in layers, allowing us to achieve independent control of important features. Eyes, brows, mouth and body can each be rendered separately so that the character can raise its brows while speaking, speak while gesturing, shift its gaze and weight when attending to user actions, etc. This capability introduces a new problem - registering the 2D layers, in particular, the head and body. In 3D, as the gesticulating body rotates around, the 2D perspective projection of the head-body registration point changes. In 2D, the separately-rendered head positions

remain stationary. Because the camera is fixed for all animations, we determine a projection matrix, apply it to the 3D registration point, and then compute a translation that aligns the 2D registration point on the head with the one on the body.

3.2 Creating Behaviors

Behaviors can thus be created to meet the evolving needs of the agent. We begin with a basic library that makes the character come alive: weight and gaze shifts, saccades, 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. Maya animation sequences are created and rendered by Cambridge Animation System's Swiffworks shader at base body orientations (e.g., toward the user, toward the screen) to Macromedia's Flash format, which provides indexing and editing capabilities. For additional flexibility, we use scalable vector graphics (SVG) as our target display format. This allows us to scale the images to the optimal size for the screen layout and employ a variety of cameras, from full body shots for gestures to closeups for conversational behaviors. To avoid the latency currently associated with parsing a document object model, we convert the final SVG files to the portable network graphics format, gaining speed but losing the ability to further scale animations.

Beyond the basic behavior library, the animators must create behaviors related to a particular domain that will be used by the character. We collected and studied video footage of human presenters, and human tutors working with users of computer applications. The reference footage permits the animators to anticipate what kinds of moves will be indispensable. In the presentation domain deictic gestures combined with beats are the most common gestures. Deictics are easy to define procedurally and generically useful. Future additions to a presentation puppet might include tutorial gestures related to orienting the user and using the application interface.

The image libraries are loaded into the puppet animation engine. XML descriptions of gestures, comprised of sequenced and procedural frames, are generated by a graphical animation editing tool. Table 1 shows the output of the editor. Background animations are performed automatically when the engine is idling, and sending a text string automatically results in lip synced text-to-speech synthesis at any of several different head positions.

4 Animation Authoring

Digital puppets require several types of input: Users must supply text for the puppet to speak and environments must supply contextual hints that enables an inferencing module to select appropriate behaviors. The selection is informed by video analysis, presentation techniques [1], and facial communication displays [8]. Based on the input, the inferencing module generates a runtime script and sends it to the animation engine for display.

Table 1. A viseme sequence and a body gesture example in Widget.xml.

```
<head>
  <!-- Frames for viseme sequence for position 000 -->
  <frame filename="head/head-000-phonemes{0001..0012}.svgz">
    <mount name="body" coordinate="0, 25.632, 0.351" />
  </frame> ...
</head>
<body>
  <!-- Frames for body gesture weight shift -->
  <gesture name="weight-shift">
    <sequence subtype="middle-to-left-to-right"
      progression="weight-shift0001..0009" />
  </gesture>
</body>
```

We use contextual hints to select behaviors in two ways. In every domain, there are naturally occurring high-concept goals. For example, in the presentation domain there are the components of the presentation task, *introduce*, *explain*, *etc.* and in the tutoring domain there is the flow of the application task. The markup for scripting the puppet's behavior at this level is shown in Table 2. The *PreIntro* is the initial context; in this state the camera might zoom in and the puppet might look around and speak. When the viewer begins the presentation, the puppet moves to the *Introduction* state, which specifies that the puppet first glance at the button, and then orient itself with its body facing toward the viewer before speaking. There are idle and end states as well. A type of *choices* will choose one [group] of the gestures, while a type of *gspeak* will specify a facial animation or gesture to perform while speaking.

The more interesting mapping of behaviors occurs at lower levels, where gestures are mapped to contexts and goals, for example, *elaborate* in the presentation domain and *increase politeness* in the tutoring domain, as well as to sentence fragments, words, and patterns, such as exclamations and numbers. This intention-based mapping is similar to the approach of Andre and Rist [3], who use presentation goals to select their animation schema. A gesture is executed randomly and probabilistically after the sentence starts. Several examples are shown in Table 3.

The animation engine executes the script commands asynchronously and in parallel, depending on the command. The script uses a special-purpose XML schema that is narrowly defined for digital puppets and was not created as a general standard. Though the 2D puppet schema is compatible with other standards, most animation languages – Affective Presentation Markup Language (APML), Multimodal Utterance Representation Markup Language (MURML), Rich Representation Language (RRL), Virtual Human Markup Language (VHML) – control 3D avatar movement and communication in virtual environments. Similarly, there are 3D facial animation standards that focus on realistic control.

Table 2. The mapping of high-level contextual information to 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 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> </c> </s> <s type=wait for=second value=5> <s type=jump context=PreIntro> </context> </pre>	<pre> <context name=Intro> <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> <i>insert authored text</i> </insert_point> </context> </pre>
--	---

Table 3. The mapping of low-level text and contextual information to 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 = "talkgesture" subtype = "default" probability = "100" /> </sentence> </pre>	<pre> <context> <entry context = "Introduction" gesture = "talkgesture1" subtype = "default" probability = "50" /> </context> <goal> <entry goal = "clain" 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, a 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.

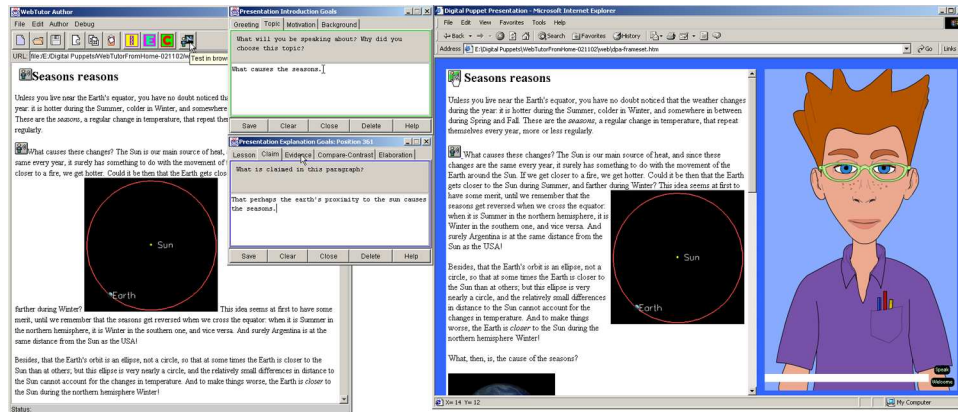


Fig. 1. On the left is the WYSIWYG puppet authoring tool with a Web page and two authoring boxes displayed. On the right is the resulting browser-based puppet presentation. Interactive buttons have been inserted in the Web page.

Our approach to authoring the puppets was based on Rhetorical Structure Theory (RST) [18] and then generalized to accommodate any presentation goal. By constraining the domain, we can reasonably infer the intent of the author with respect to the authored commentary. For example, in the introduction, a user must provide text to motivate the topic. Based on the context and goal, the inferencing module might select behaviors appropriate to motivation and introductions, such as face viewer, perform inviting gesture, and amplify voice.

Figure 1, left, shows the digital puppet user authoring tool, which contains a Web page about the topic to be presented. Clicking on a paragraph of text brings up a small authoring windows, shown at center, that displays the particular concepts that the students must address in their explanations. These goals (e.g., *lesson, claim, evidence*) can be tailored for different contexts (i.e., *introduction, explanation, and conclusion*.) Students provide an introduction, a conclusion and several paragraph-linked explanations that, when played together, will result in

an oral presentation about the subject matter. 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), the 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 Figure 1 and Figure 2. The user activates the presentation by pressing the interactive buttons.



Fig. 2. Skip gives a presentation about digital puppets at an AIED conference event.

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 within the Virtual Factory Teaching System, an educational factory modeling application and simulation to teach undergraduate engineering students production and inventory management. (See Figure 3.) If the expert chooses to initiate a speech act, in order to comment or give advice, he selects an appropriate topic and politeness level from the expert 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 [5], 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 provides 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.

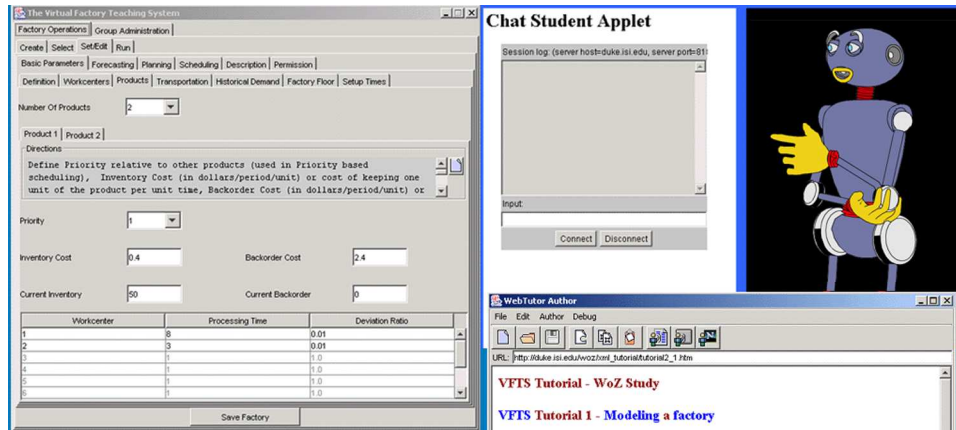


Fig. 3. Widget tutors a student using the the Virtual Factory Teaching System.

6 Conclusion

Digital puppets have potential as online commentators and presenters because they combine ease of authoring and automated behavior generation, allowing novice users to build characters that can act with limited autonomy within their particular domains. By generating 2D puppets from 3D characters, we can quickly create new animations and build up libraries of behavior. We have deployed two different characters in two different environments - presentation and tutorial - and have found the process of moving to a new domain to be fairly efficient. As computing power increases and characters can be animated more quickly, believability will improve and conversational interfaces will become an interface standard. Because reasoning in complex domains will remain a hard problem for much time to come, we expect limited autonomous agents such as the digital puppet to bridge the gap between narrators and intelligent tutors.

References

1. Gross Davis, B.: Tools for Teaching. Jossey-Bass. (1993)
2. André, E.: The Generation of Multimedia Presentations. (1982)
3. André, E., Rist, T., 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)

4. Ball, L.: Lifelike computer characters: The persona project at Microsoft (1996)
5. Brown, P., Levinson, C.: Politeness: Some universals in language use. New York: Cambridge University Press. (1987)
6. Cassell, J., Vilhjalmsón, H.: Fully Embodied Conversational Avatars: Making Communicative Behaviors Autonomous. *Autonomous Agents and Multi-Agent Systems* 2(1): 45-64. (1999)
7. Cassell, J., Vilhjalmsón, H., Bickmore, T.: BEAT: the Behavior Expression Animation Toolkit. *SIGGRAPH Computer Graphics Proceedings*, AMC Press (2001)
8. Chovil, N.: Discourse-oriented facial displays in conversation. *Research on Language and Social Interaction*. (1991)
9. Conversive: InstantAgents, <http://www.conversive.com>
11. EDrama Learning: Natural language in simulated conversation. (2002)
12. Ganeshan, R., Johnson, W.L., Shaw, E., Wood, B.P.: Tutoring Diagnostic Problem Solving. In *Proceedings of the Fifth Int'l Conf. on Int. Tutoring Systems*. (2000)
13. Haptik: Automated Personalities, <http://www.haptik.com>
14. Johnson, W.L.: Using agent technology to improve the quality of Web-based education. In N. Zhong & J. Liu (Eds.) *Web Intelligence*. Berlin: Springer. (2002)
15. Johnson, W.L., Rickel, J.W., 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)
16. Lepper, M.R., Henderlong, J.: 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. (2000)
17. Lester, J.C., Converse, S.A., Kahler, S.E., Barlow, S.T., Stone, B.A., Bhogal, R.S.: The persona effect: Affective impact of animated pedagogical agents. In *Proceedings of CHI '97*, 359-366. (1997)
18. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, ISI, Univ. of Southern Cal. (1987)
19. Marsella, S., Johnson, W.L., LaBore, C.M.: Interactive Pedagogical Drama. In *Proceedings of the 4th Int'l Conference on Autonomous Agents*, Agents 2000
20. Marsella, S., Johnson, W.L., LaBore, C.M.: Interactive Pedagogical Drama for Health Interventions. In *11th Int'l Conf. on AI in Education*. (2003)
21. Oddcast: SitePal, <http://vhost.oddcast.com> (1996)
22. 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)
23. Reeves, B. and Nass, C.: *The media equation*. Cambridge University Press. (1996)
24. Shaw, E., Ganeshan, R., Johnson, W.L., 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)
25. Shaw, E., Johnson, W.L., Ganeshan, R.: Pedagogical agents on the Web. In *Proceedings of the Third International Conference on Autonomous Agents*. (1999)
26. Stone, Lester: Dynamically sequencing an animated pedagogical agent. (1996)
27. Rist, T., André, E., and Müller, 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)
28. Rizzo, P., Shaw, E., Johnson, W.L.: An Agent that Helps Children to Author Rhetorically-Structured Digital Puppet Presentations. In *Proceedings of ITS*. (2002)