Authoring Edutainment Stories for Online Players (AESOP): A Generator for Pedagogically Oriented Interactive Dramas

Barry G. Silverman, Michael Johns, Ransom Weaver, Joshua Mosley
Ackoff Center for Advancement of Systems Approaches (ACASA)
University of Pennsylvania, Philadelphia, PA 19104-6315
barryg@seas.upenn.edu

ABSTRACT

The video gaming industry has experienced extraordinary technological growth in the recent past, causing a boom in both in the quality and revenue of these games. Educational games, on the other hand, have lagged behind this trend, as their creation presents major creative and pedagogical challenges in addition to technological ones. By providing the technological advances of the entertainment genres in a coherent, accessible format to teams of educators, and developing an interactive drama generator, we believe that the full potential of educational games can be realized. Sections 1 and 2 postulate three goals for reaching that objective: a toolset for interactive drama authoring, ways to insulate authors from game engines, and reusable digital casts to facilitate composability. Sections 3 and 4 present progress on those tools and a case study that made use of the resulting toolset to create an interactive drama. We close with lessons learned and challenges remaining.

Keywords: videogame generator, role playing games, interactive drama, training, stealth learning, agent approach

1. Introduction and Goals

We envision a future where many games exist that help people to cope with their health issues, child rearing difficulties, and interpersonal traumas. Further, these games will be so compelling and easy to revise, that many players will feel compelled to contribute their own story to the immersive world – a contribution that is both self-therapeutic and that helps others who see some of their own dilemma in that story. This will be an industry that is consumer grown, since they will be the creators of new games for other consumers. As a few of many possible examples (1) parents will experience what other parents of handicapped children have struggled with and overcome, (2) children who are bullies will learn what their bullying does to other kids, and (3) people with chronic health issues (over-eating, diabetes, heart disease, etc.) will learn what happens when self-denial and poor diets prevail. We envision that a single underlying game editing environment and alterable cast of digital characters can be used to facilitate such a variety of games with therapeutic value.

At present there are many obstacles to this vision: (1) the videogame industry offers addictive, immersive entertainment and provides most of the seeds for this industry to grow from, however, their games have little education focus and they provide few if any tools directly re-usable in this niche; (2) the computer-based education field does produce interactive training tools, however, these are heavily corporate and government training based and have almost no entertainment value and hence aren't spontaneously fueling much consumer interest; (3) the field of movies and TV show writing creates compelling characters that consumers care deeply about, but this medium offers no chance of interactivity that is vital to self-discovery and skill development; (4) the field of human behavior modeling offers innumerable models based on first principles of physiology and psycho-social dynamics, yet outside of a few experimental military simulators, these are rarely inserted into autonomous characters in videogames and interactive dramas; and (5) the successful edutainment offerings to date (e.g., Math Blaster, Reader Rabbit, Oregon Trail, etc.) are monolithic, non-alterable creations of their proprietors. We need a next generation of environments that takes the best from each of these fields and provides the needed capability. The elements of this environment mostly exist, but they haven't been properly put together yet.

We believe one could take the important elements that exist today and synthesize them into the desired capability for Authoring Edutainment Stories for Online Players (AESOP). Provided the game authoring toolbox (what we call AESOP) is usable and useful, then game-authors will be able to 'write about' their situations and game-players will benefit from immersively seeing and experiencing the problems that others have had to deal with. **The first goal** of this research is thus to explore ways for a game generator to help authors introduce entertainment and free play into role playing games and interactive dramas that are training interventions.

This goal is compounded since learner-oriented game designs are one of the most difficult areas in developing videogames. First off, although training requires players to progress through stories (pedagogically valuable scenarios), at its heart game play is not about interactive fiction though there are those who buy interactive fiction games. Interactive drama is all about storytelling from the author, while game-play is much more about story creation by the player – and these competing aesthetics need to be resolved if pedagogical games are to achieve their potential in general.

More than any other mechanic of game-play, narrative in game raises the idea of destroying the central aesthetic – that players create their own stories and that is what keeps them coming back. Other game-play mechanics more or less have a story inside them, in fact countless stories inside them. Further, many of these mechanics have built in skill training functions at the same time that they permit unconstrained play and inquiry. For example, a racing and chasing mechanic includes lessons about how to chase down bad guys and cut them off from escape. If one invests in the realism of these mechanics, they provide useful training and transferable skills [3]. The same should be true if interactive dramas are well done, particularly if the goal of the drama is to learn, rehearse, and transfer skills in interacting with people; and/or to learn how to persuade people to change dysfunctional behaviors and by that to learn how to cope with one's own poor health behaviors before they become a real world problem. That is, dramas are essentially dialog games, and hence one must take precautions to preserve the gameplay aesthetic.

It's also a fact that students learn the most and retain it the longest when they must teach a topic to others. One always learns a subject better if one is confronted with being the teacher rather than the student. So a microworld could be quite a powerful training device if it if it thrusts the player into roles where other characters will be vulnerable and dependent on the player to teach for a successful conclusion to be reached. Why should the player care to become a teacher? What can drive them to reach this level of learning? People reflect this kind of passion for videogames and at the movies. When game mechanics work and when characters are likable, players (viewers) achieve enormous empathy for the characters and are willing to go to great lengths to save them and to help them work out their problems (e.g., as in 'God' games such as The SIMS and Tamagotchi), and to go on quests on their behalf or assist them in shifting their behaviors to more successful models such as in role playing games. In these milieus, players reveal willingness to learn skills that will help the characters.

At this point, let us restate **the first goal** as researching a generator that permits authors to create interactive role playing games that preserve the central aesthetic of gameplay, that utilize stealth learning and self-discovery in microworlds for training and behavior change purposes, and that incorporate learning by teaching. A **second goal** is to provide a high level graphical user interface for the generator, and by that to insulate authors from having to learn a game engine's details.

2. Creating Stories With Free Play

As already mentioned, we are seeking to set up a generator that can expose constructs and parameters of a storyworld so that new interventions may be more readily authored that promote free play and entertainment within a narrative structure. To support this research, we are attempting to produce a cast of animated puppets and sets (introduced in what follows) in a way that they can be reused for many stories (**third goal**). This is the idea of a composable and reusable storyworld, including digital sets, cast members, and Campbellian archetypes that can be adapted, and extended for further sequels not even yet anticipated. Our ideas for reusable casts and archetypes follow from work such as [1, 2, 11] as well as how they are used in franchise games, comics, and serials. We include characters of different ages, genders, and backgrounds/ethnicities, and in the roles of hero, sidekick, allies, opponents, tricksters, lovers, and so on.

It is worth pointing out that, for now, we made a conscious decision to base this cast and sets around 2-D, hard-edged cel-based animations since research has shown that subjects with health behavior change issues often allocate little cognitive processing to health messages, and feel greater confidence about being able to process and conquer message sets introduced in cartoon formats [4]. However, the underlying technology also supports 3-D animations, as is used in our Unreal Tournament version for military training.

In addition, we chose a finite state machine (FSM) approach as the basis for our dialog model and our scriptwriting application. The FSMs may be represented visually within a directed graph or tree. Edges represent the various dialog choices available to the user after a given node plays out. Each node contains both dialog and animation instructions for the avatar and Non-Player Characters (NPCs) to carry out and that may be activated in parallel. This approach allowed our writers to choreograph the animation of multiple characters to occur simultaneously. The AESOP generator is currently implemented to help authors with the FSM approach and so that it can encapsulate and deliver the interactive game to other devices that display and track game play. Section 3 will explain this structure in more detail. Before that, however, it is important to further explore how AESOP seeks to satisfy the first design goal.

2.1. Narrative Intelligence

The field that some refer to as 'narrative intelligence' has recently produced a number of rich ideas for incorporating narrative into game worlds without totally sacrificing gaming's central aesthetic. In this research, we synthesize, adapt, and extend several of these ideas as this section will note. None of the literature to date has directly addressed the topic of learning by teaching, so this places us in a new realm that drives our inquiry. Also, very little if any of the narrative intelligence research to date addresses how to assist storyworld authors, so several original contributions are needed here to realize our goals. We are creating the AESOP generator as part of this research and are seeking to have it assist with authoring constructs as in Figure 1 and as further described in what follows.

From the player's perspective, when they encounter a storyworld such as in Figure 1a, they do not wish to be placed on 'rails' -- a storyline forcing the player down a narrow path that is author specified. The best narrative solutions approach this concern by interspersing free play/inquiry and player story-creation with player-selected choice points for advancing the story. At these choice points, the player approaches characters or other devices that reveal more of the author's story and that advance them to the next scene of the drama. In this manner, a drama eventually unfolds. Some successful examples of this blending of story and game are Grand Theft Auto, Deus Ex, and America's Army [12].

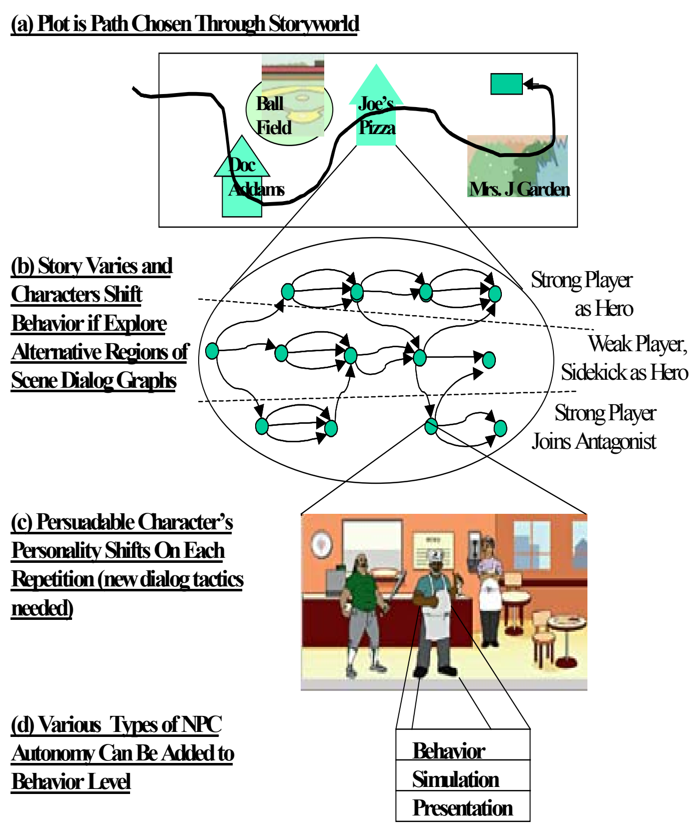
Similar to this is the approach being taken in the Mission Rehearsal Environment [16], however, unlike the popular titles this approach is for doctrinally correct training. Its approach requires authors to (1) deconstruct the story into the smallest parts (scene nodes) where autonomous character and player freeplay can be permitted, and (2) to identify graph transitions that are gate conditions for triggering scenes and/or for allowing scenes to be omitted without loss of training value. This approach permits the player to explore a node repeatedly, getting better with each try and through exit node feedback. However, this approach is for doctrinal training that requires repetitions for improvement.

In the current research we are interested in learning by teaching, in mental model transfer, and in behavior shifting as mentioned earlier. This relaxes the need to repeat the identical scene, and affords the opportunity for scenes to hold surprising plot reversals if you play them differently. In storytelling theory, the listeners, or participants, are viewed as containing significant understanding and know-how already, and the story is but a fuse to ignite the recipients into synthesizing a new conceptualization for themselves. This is also consistent with the Persuasion Likelihood Model [10] which states that rational arguments are unlikely to persuade. Rather it is the peripheral cues that are modeled which convince the audience. Thus the many movie scenes of actors smoking during or after a significant activity have far more power than all the public health media campaigns laying out arguments about adverse health effects.

Figure 1b zooms us in on a plot or dialog graph for a sample scene or quest of the storyworld. Here the nodes and edges are as described for the FSM in the prior section. Depending on the player's personal goals, entertainment objectives, style of play, and confidence, among other factors, they may decide to pursue very different avenues through a scene's dialog graph and in fact through the entire story world. This can be both educational and entertaining. Monkey Island is an example of a title that incorporates argumentation tactics in this manner, though in that title there is only one outcome and path out of a scene regardless of your dialog choices. Popular games like Civilization, Black & White, and EverQuest, in turn, have richer outcome possibilities and they let the player learn about the world's emergent nature and how they have to live with the worlds they create and the personas they project.

The same types of explorations can exist in pedagogically oriented dialog graphs. Earlier efforts introduce such possibilities by providing side characters that coach and cajole a player back to the pedagogically preferred path [8, 14] of a dialog graph. In the types of storyworlds we currently envision, however, the learning can be just as effective if a sidekick or "window character" regrets aloud the player's decisions and then on its own directly performs the dysfunctional character training and persuading [2]. In theory, the player should even be able to adopt a potential storyworld antagonist's causes and be entertained by helping to support the antagonist's objectives throughout the storyworld, yet suffer no loss of learning as a result. We are finding a lot

Figure 1 – Sample Insertion Points for Story Writers to Minimize Perceptions of Limiting Free Play



of success with including such dialog plot regions and archetype roles for cast members to readily fill in, particularly when players grow bored late in a training-oriented game (see Case Study of this paper and [17]).

There have been a number of investigations into conversations with autonomous agents outside of stories and/or in fixed plot graphs. Some of the earliest work dealt only with simple animation and kinesthetic issues such as breathing and blinking, lip synching, and facial expressions – things we label as "presentation layer" in Figure 1d [e.g., see 6]. More intriguing, however, is work on the higher layers of Figure 1d and how they might be integrated into dialog plots as suggested in Figure 1c. We define the Simulation Layer of an agent as how it performs in the world – e.g., navigation, collision avoidance and damage, and physiologic and health needs. In some characters, we embedded a number of validated reservoir models of body organs and functions [15]. Less realistic models are widely used in popular God games, however, there is rarely any story designed into it, yet players ascribe story when uncorrelated events arise. In our work these eventually are intended to provide potentially engaging dialog opportunities, particularly with characters that attempt to deny or mislabel their symptoms, risk factors, and habits.

Likewise, the Behavior Layer involves characters' emotions and motivations, planning/choosing style, and general personality variables including coping modes. One idea here is to allow NPC or non-player characters' moods and personalities to respond dynamically and emergently to direct player interaction. The NPCs include parameterized models of autonomous, emotive behavior and different types of responses to player actions or dialogs. The Virtual Theatre Project [5] has explored this concept for fixed plot graphs and shown that players perceive significant dramatic variability and story-creating potential, even though the plot is fixed. An early prototype of Heart Sense Game has likewise deployed an autonomous coach/companion that alters its mood, emotion-directed utterances, and physical expressions as a function of where the player strays in the plot or dialog graph and found this reduces player difficulties [14]. In the current research we have eliminated overt coaching, but are attempting to use this idea for key characters so they alter their personality each time you play. This in turn furthers the perceived player variability and sense of free play.

Silverman, Johns, et al. [15] demonstrate how a number of models from the literature across these two layers (simulation and behavior) can contribute to making agents autonomous and need-reservoir driven in their coping styles and emotive decision making. This is an attempt to move beyond Bates' believable and broad agents [9] into the realm of reliable models of human performance calibrated against field data – an area where learning systems must depart from entertainment.

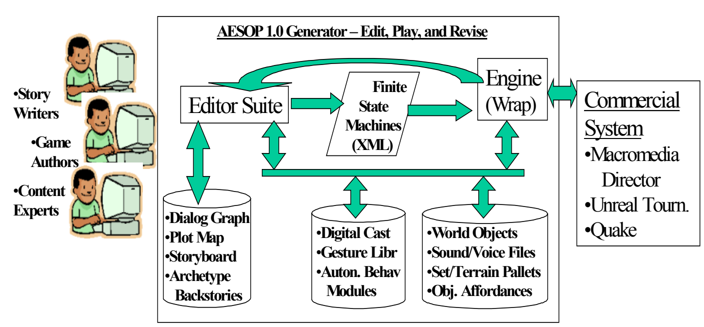
A final issue facing storytelling in simulated worlds is that it forces the player into what is arguably the worst side of human-computer interaction, that of the computer's poor conversational capabilities. As with voice menu systems on the telephone, the machine-generated voices are stilted, their ability to handle nuances are poor, and they often misunderstand the speaker. Up to now we use a text to speech system during authoring but replace it with actor voice-overs once the parts are finalized. We completely avoid the speech recognizers and instead rely on dialog menus which raises several difficulties. Specifically, the risk of dialog menus is that the designer has neglected to include options the player would like to see voiced, or if they are voiced, hasn't included mechanics in the other characters to support the idea in the player's head. So far in the case study, however, we have not encountered this difficulty and believe the large degree of free play mentioned in this section tends to minimize the dialog menu risks.

3) The AESOP Generator

AESOP is intended as a front end authoring aid that includes plot and dialog editing GUIs (graph markup language), storyworld templates, pallets of reusable parts, digital cast members, autonomous behavior modules, and reusable art/animation assets. It's output is automatically parsed into XML instructions for each agent in the storyworld in the form of finite state machines (FSMs) that are sent to the game engine. With the use of an XML interface, the AESOP editor suite becomes engine independent. Its FSMs could in theory be played by any of a variety of game engines that run NPCs and avatars. Figure 2 overviews that architecture, and the discussion that follows provides further details.

When building a piece of edutainment, the generator must support the entire group including training content developers, story writers, and game authors. A goal of this research was to study one or two

Figure 2 – Architecture of the AESOP Generator to Insulate Commercial Engines and Help Authors Create Interactive Dramas



such groups as they attempted to create an edutainment system, elicit their design protocols and intermediate game representations, and to try and craft a generator environment that might better support their mutual and collaborative efforts. In the latter, we were hopeful of placing various tools and versions of a generator in front of them to further the requirements observation and elicitation process and to study environment design concerns. Where and when we did not have a specifically needed tool, we intended to support the need manually, by directly programming the authoring need and/or game mechanic. In this fashion we are engaging in a spiral development of the AESOP generator.

In terms of specifics, the lead author of this paper serves as principal investigator of both the AESOP generator [17] and two edutainment projects that are making use of it -- the Heart Sense Game (HSG) role playing drama [14] and a recreation of Mogadishu/Black Hawk Down (BHD) crowd scenes for a first person shooter scenario [13]. The former of these applications connects AESOP to an engine written in Director, while the latter connects AESOP to the Unreal Tournament game engine. Also, a third project called Athena's Prism is just getting started that will utilize portions of the AESOP environment as well, but which will invoke a locally produced game engine. This article focuses primarily on the HSG and Director version of the AESOP generator; though discussion at times will mention features and lessons of the other applications.

The HSG started last quarter 2001, and since that time the HSG development group met at times weekly and at other times bi-monthly for 90 minute face-to-face discussions. This group included 6 faculty investigators, 2 grad student researchers, at times up to 9 undergraduate digital media design and systems engineering students (helping with art, animation, sound, voiceovers, etc.), and 1 junior and 1 senior screen writer (part-time, freelance). They were supported in between these meetings via a variety of collaborative tools including threaded chat, web-based ftp repository (organized into sub-team memory bins), email listserv, and general email. Threaded chat was highly useful at the outset for discussing learning objectives and game mechanics, but, as the threaded conversations grew, people found them cumbersome and resorted to email and direct meeting instead. FTP repositories followed a similar pattern, although there is also some use of memory sticks, CDs, and other media for exchange.

Figure 2 shows two boxes labeled Editor Suite and Engine. Various tools were placed into these boxes and evaluated/improved over time, as subsequent sections of this article suggest. As an overview of that discussion, the plot map (acts, scenes, etc) and character backstories started as text-only descriptions, evolved to a manually filled in multimedia set of webpages (http://www.seas.upenn.edu/~barryg/heart/index.html), and is now targeted to become an interactive editor that will assist in merging learning objectives with story writing goals. The earliest versions of the branching, interactive dialog script were table-based which was then replaced by a directed-graph editor (Section 3.1). The earliest versions of the game engine existed prior to the artwork/Flash movie stores and utilized stick figures (with text to speech) to act out the roles and dialogs from the script. Subsequent versions included a library of sets and characters replete with growing stores of gestures (Flash movies) and actions one can assign with a mouse-click to the character puppets (see Section 3.2). One authors dialogs and actions in the graph editor tool. This produces scripts in text and graph markup language (GML or XML) that are instruction sets or finite state machines that the Engine can run atop Director with the help of a text to speech (TTS) processor and the library of Flash movies for each character's gestures and actions. Thus there is no need to program in Director, and developers author role playing dialog scenes and watch them acted out with the push of a button, provided they aren't expecting gestures and actions that aren't yet in the Flash movie stores for each character.

At times we have included autonomous emotive agents in earlier versions of HSG, agents capable of emergent behavior [14, 15], while the BHD and Athena's Prism make substantial use of such autonomy. These are NPC agents that operate with their own behavior goals, standards, and preferences, and that can react to and effect the drama and the player. The current article omits discussing these characteristics, but we have numerous papers on this topic: e.g., see [15, 17] among others, and we continue to work on the challenges of integrating author- vs. agent-driven story elements.

3.1) GraphEdit Tool

Our finite state machine editor is a modified version of Visual Graphs for Java, developed at Auburn University. To facilitate our particular needs, the second author of this paper added custom dialog boxes for the data we manipulate, and added support for the XML output required by our game engine.

In our graphs, nodes contain uninterruptible segments of storytelling, and edges correspond to the choices given to the user after each node plays out. Within each node is a set of behaviors assigned to various characters, arranged as a tree. Behaviors can be any of the following: 1) *Speak*, which causes the character to lip-synch a line of text either defined by a .wav file or, failing that, a text-to-speech generator; 2) *Gesture,* which causes the character to perform some specific animation; or 3) *Move,* which causes the character to physically move from one position on the screen to another. When one behavior finishes, all of its direct children are executed in parallel. This allows for authors to specify the timing of various components of a scene without knowing specific details about the art or voice assets that will eventually be put in place.

Once the tool is utilized, one can save the graph out to XML format which is then passed to a subsequent module for parsing and linking with the game engine. A sample of the XML output for the node shown in Figure 3 includes:


```
<ACTION entity="Jack" starttime="166846" endtime="15394585" comment="" target="" action="Janus_RightChestRub" framespersecond="24" volume="125" />

<ACTION entity="Joe" starttime="12313825" endtime="166846" comment="" target="" action="Joe_HoldStomach" framespersecond="24" volume="125" />

<SOUND entity="Wanda" starttime="250536" endtime="12313825" comment="" target="" filename="Wanda_4106.wav" subtitle="She's right. We've got to get you to the hospital." voice="Mary" showvoicebubble="True" volume="125" />

<ACTION entity="Wanda" starttime="250536" endtime="3574877" comment="" target="" action="Wanda_Left_Gesture1" framespersecond="24" volume="125" />

<SOUND entity="Sheila" starttime="7694236" endtime="250536" comment="" target="" filename="Sheila_2534.wav" subtitle="Joe, don't stall-- make the call!" voice="Mary" showvoicebubble="True" volume="125" />

<ACTION entity="Sheila" starttime="7694236" endtime="16440838" comment="" target="" action="Sheila_HandAntacid" framespersecond="24" volume="125" />
```

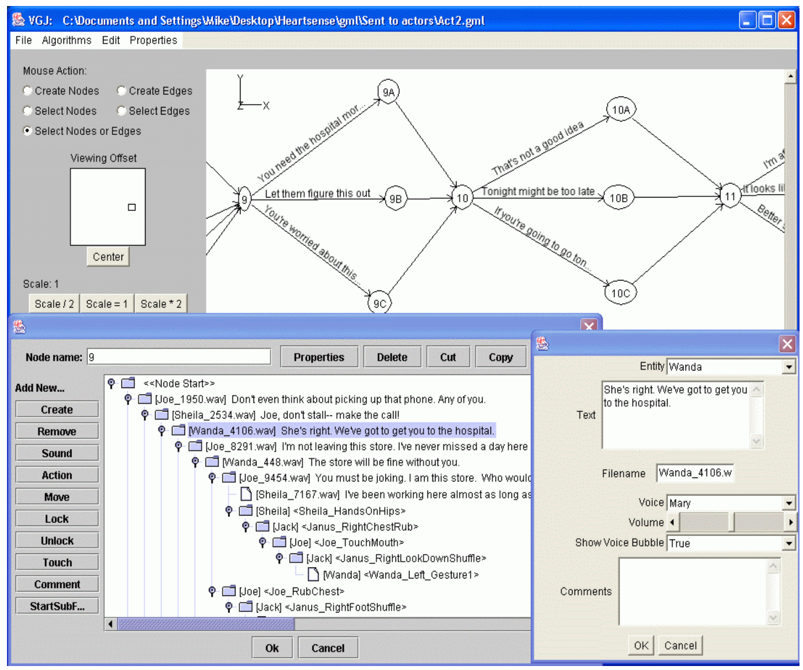

3.2) Gesture Builder

The fourth author of this paper, and his Digital Media Design and Fine Arts students, have created all the artwork for the reusable casts as well as the sets and terrain objects for the HSG version of AESOP. The Flash artwork was developed in tandem with the story development using a stylus pen and Adobe Illustrator. Each body part was drawn on a separate layer to aid the construction of the micro Flash animations. To provide the maximum flexibility, it was essential to build the animations so they could be run independently and simultaneously.

In addition, they created a Director-based Macro Animation Editor shown partly in Figure 4, in order to test how the Flash animation segments would flow into each other and in order to build gestures, or encapsulate sequences of animation that could be utilized by the scriptwriters. A Gesture, such as "look angry" might include a change of facial expression, a shift of weight and movement of arms, forearms, and hands so they raise to the hips. The resulting coordination of this animation would be exported as a Gesture. For more common animations, such as "walk left", the animators would build an eigtht-frame looping walking sequence that could be called by the macro editor, by rotating feet, shins and thighs.

This structural approach is fairly common to the game industry, however, in this particular case it was necessary to provide a simple interface and upgradeable characters that could accept new animations on a need-by-need basis as the story development team authored stage direction. Motion capture, and post-capture editing, would be another technique of generating animation fragments that could integrated with this Macro editor.

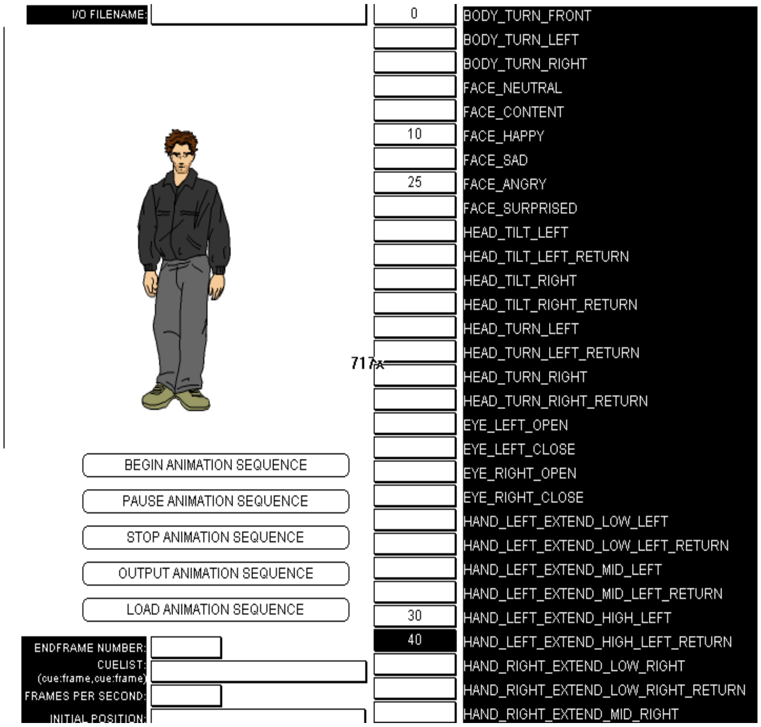
Figure 3 – Select Screens of the GraphEdit Tool for Branching Dialogs and for Adding Choreographic and MultiMedia Instructions to FSMs



3.3) Engine/Wrapper

In order to get Macromedia Director to make use of the FSMs in XML, the third author of this article created a game engine in Director's script language, Lingo. The goal was to insulate authors from Director syntax and instructions, as already mentioned. Director makes use of a stage metaphor and expects inputs in the form of a "score" that holds instructions for sprites to set stage backgrounds and to bring potentially autonomous props/puppets/artifacts onstage or take them offstage according to timeline triggers. The score may include tracks so that parallel activities can be supported such as, for example, background sounds (track 3) with foreground voiceovers (track 2) as the puppets are lip synching and carrying out various gestures and motions (track 1).

Figure 4 -- Overview of the Flash Macro Editor for Building Gesture Procedures and Movies



The Lingo Game Engine we created is an algorithm that essentially parses the FSM markups or instruction sets, and translates them into Director understandable Score syntax. It also assures program coherence, eliminates track conflicts, enforces agent/object turn taking, assigns resources and procedures from libraries (voice files, animation movies, etc.) to puppets, and handles input from the user. As previously discussed, the behaviors assigned to characters within a given node are arranged in a tree, with direct children executed in parallel upon completion of parents. The role of the engine, then, is to examine this tree of behaviors, determine whether any currently executing behaviors have completed in the last frame, and if so begin its children. When all behaviors in a node have completed, the engine presents all outgoing edges as choices to the user. When one is selected, the next node begins.

4. Results To Date

The AESOP generator has been developed in parallel with the creation of the HSG, and with the goal of supporting the authoring of that game. No game is likely to succeed if it's appeal cannot be summarized in few a sentences. To initiate the process, the first author of this paper came up with a short description of the intended game and a paper-based version. Specifically, for HSG, this description is (after some massaging from HSG co-investigators): *Heart Sense Game is a role playing game in which you help the hero try to solve a crime and simultaneously rescue his career and find romance. However, as the hero, some of the many characters you might get clues from, need your help to deal with heart attacks before they or others can help you. Since, for their own reasons, they often don't believe they are having a heart attack or don't want to take care of it promptly, there are significant obstacles to helping these characters to help themselves. And if you prefer to harm these characters, you are free to do so, but watch out, your own future will be effected as well!*

The three act, character-driven soap or adventure story is a proven formula both in the movies and on TV. The writer's immediately recognized this format and could relate to its conventions to drive the player and his or her avatar through the story summarized above. Likewise the training content developers could identify with a hero's journey. Jointly, writers and content experts began to make passes over the story to preserve its engagement aesthetic. The various authors provide brainstorming ideas and interactively deepen the script. The writers tend to form narrative descriptions of the scenes and the training developers begin to allocate their learning objectives to these quests and scenes. A negotiation goes on where the dramatically inclined attempt to limit the learning objectives in any given scene, while the trainers try to assure their full set of goals is covered somewhere in the overall journey.

The extent of the training objectives determines in part the length of the story, and the number of quests that must be included. Thus for example, in the heart attack domain there are multiple types of heart attack presentations, and three main categories of behavioral delay. After brainstorming, the goal of limiting the length to that of a TV show (1 hour for once through) eventually ruled the day and limited the journey to three quests in total during Act 2. Further, writers insisted that each scene should move along quickly and not sacrifice dramatic pace for the sake of training detail. Since persuasion theory supports this idea (peripheral cues are of high value), the negotiation landed on the side of less-is-more. In either gaming or storytelling, its vital that each line of dialog potentially has three purposes: move the plot along, reveal some aspect of the speaker's backstory, and set up any local effect (e.g., joke, action, lesson, etc.). To make this happen, each character in a scene needs to have a well defined, story-pulling role, and this required a number of discussions and rewrites about dialogs, plots, scenes, and beats.

A problem with the storytellers' design was that the result was linear and largely non-interactive. While they are skilled at bringing in tension, climax, and drama, writers tend to do so by placing the human on 'rails' and in a passive listening role. To facilitate a shift in this mindset [7] and to help illustrate the nature of the game on the computer, the authors and content developers were shown a concise version of the game on note cards so that they could play it and try out the game. Flipping through note cards should allow players to experience the opening scene (Act 1 note card), the various quests in Act 2 (1 note card each), any reversal of plot (flip side of each note card), and what happens to the hero and other characters along the way and at the end of the game (Act 3 note card indicating possible outcomes).

There are several central aesthetics and mechanics that should emerge from this and subsequent deepenings of the game design. For one thing the designer tries to preserve the overall entertainment score (ENT) which is a function of the (F) or "fun quotient" that derives heavily from four contributing sub-processes, at a minimum, including: (R) "Rules" of the game that are satisfying such as how to move around, interact, fight, persuade, etc.; (C) "Control" remains in the hands of the player in the sense of creating his own story and selecting tactics to deal with dilemmas along the way; (O) fairness of outcomes along the way and in the end (did player's choices have believable consequences) - social contract between designer and player; and (AT) in which the game provides the player with an opportunity to **a**ccumulate things, **t**hreatens him with their loss (scare thrill), allows him to try and protect them, etc. These four mechanics are central to almost any game, and they must be appealing if the game is to be fun for the player. However, they also must be tuned to the class of player (demographics) targeted for the specific game.

After the writers and training developers worked with these ideas for a while, it became apparent that parallel story outcomes needed substance if they were to intrigue the player and offer tempting alternate goals to strive for. Players need a reason to be "BAD" in the healthcare sense. That is, the player needs an alternate but legitimate set of goals to try and achieve – another story that might make as much sense and be as much fun as the "good" path (helping to heal everyone in town). This had to be a legitimate alternate story and set of goals such that the player could only pursue it adequately by being brusque with the various health victims. In the end we found three strong storylines, and deployed them to support the idea of players controlling the outcome. Also, in each storyline there were contagonists modeling proper cues and providing feedback, rewards (things to collect like career options, family relationships, etc.) and antagonists meting out punishments or threatening things that might be lost. For each of these three main storylines, asked the writers to create several dialog strategies as for the GOOD storyline alone.

In general the writers were uncomfortable in authoring their three story versions on anything other than a word processor. Not wanting to destroy their creative processes, we supported this effort, and then had a secretary move the dialogs to the graph editing tool after they were finalized. The authors and content experts then verified the results both via printouts and via play testing. The end results included about 100 pages of script which translates into 346 state nodes, 480 edges, 691 dialog acts, and 1346 gesture commands invoking 461 unique gesture macros. Overall, this authoring effort required about 1 person-year broken down in round numbers as 500 person-hours from the dialog writers to author the script, 80 person-hours for the secretary to enter the script into the trees, 400 combined person-hours from the two graduate research assistants to add the choreography to the trees, and perhaps 6 person-months equivalent across all the faculty co-investigators (content developers, story critics, and play testers).

5. Discussion of Results and Next Steps

Our research up to this point has revealed some surprising facts. First, there are no environments one can turn to for rapid authoring of pedagogically oriented interactive drama games. While games from other genres are beginning to arrive packaged with sophisticated editing tools, the educational gaming community generally is forced to create non-modifiable games on a per-subject, per-audience basis. There exists a growing number of tried and true guidelines for creating fun games. There exists a huge body of work on the subject of effective methods of education. And narrative has its own effectivity metrics. But, at present, most games are designed from the start with entertainment as the primary goal, with any learning on the part of the player as a beneficial side-effect. Pedagogical games, on the other hand, begin with rigid learning objectives that must be satisfied, which place severe constraints on the design of the game. This tension has created a deep gap between the creators of educational games and the creators of entertainment games, and consequently little mutual benefit is generated from work in either community.

We believe that the solution to this problem lies in the creation of a system that provides the building blocks of interactive storytelling by implementing the inner workings of a variety of gaming devices as composable parts, with their actual arrangement and content determined by the educators. Dialog, character movement, puzzle manipulation, resource models, combat, and other mechanics would be weaved generically into a unified game engine, with the educators able to simply choose which ones suit the story, pedagogical goals of the game, and the needs of the target audience. While a game like HeartSense is inherently dialog-oriented with its focus on persuasion and interpersonal relationships, our Black Hawk Down recreation in Unreal Tournament is a first person shooter training game with autonomous agents and emergent crowd behaviors. Such a unified engine is becoming an increasingly realistic possibility, with many recent games beginning to blend elements from a variety of others, causing genres, and more importantly game engines, to converge. Given an environment such as this to work within, designers can harness the state of the art in the technical aspects of interactive storytelling while staying focused on content creation.

In terms of the three goals stated at the outset of this paper, there has been some forward progress, and with it has come the realization of new challenges.

Goal 1: Support the authoring of interactive pedagogical dramas – Thus far, our animated stories have been constructed initially in a word processor to permit creativity to flourish and then, once they are stable, they are transferred to a graph or tree environment. By adopting an FSM approach, this lends itself to a visual graph or tree representation and creating this is an essential intermediate step before an engine can execute the game. This is straightforward for the programmers and to a lesser extent the writers, and without a step such as this, our game never could have been built. The FSM tree representation was a vital step, but it is probably not the final resting spot for our AESOP generator. There are two major difficulties that we encountered: the writing effort before using the graph and the choreographic load once the dialogs are inserted into the graph. In terms of the first of these, the creative effort required of the writers seems substantially greater than for writing linear stories, or even three linear stories at once. Our ideal system should require no more creative effort than a linear story, but should draw this baseline story out of the writer in such a way as to allow for many degrees of interactivity. In the end the FSM tree had no noticeable impact on reducing the load for the writers. They stayed away from it, and only after we locked in their scripts and dialogs did we then use the FSM tree. Our future research is now aimed at finding tools that might actually ease the writers' burden. For example, one possible step is to eliminate unnecessary ordering constraints that a tree or graph approach imposes. For instance, in our current system, character A may ask B a question, and upon getting an answer asks C an unrelated question. While possible to allow the user to ask these questions in either order, state explosion becomes a problem quickly, leading to our writers basically not bothering. By adopting a less strictly specified design, we can avoid this problem entirely. The under-specification of the tree might allow natural opportunities for interactivity, while simultaneously lending itself to adaptation from a linear script. Authors need only to ask themselves which lines in their original story *must* come before which others, and the system could potentially take care of the rest. Another idea we think worth pursuing is to borrow from the extensive number of combat systems created by the gaming community to model certain forms of verbal conflict. Where other combat models use such concepts as ballistics and material densities, this model would turn to the rhetoric model introduced by Aristotle. Beliefs, arguments, and counterarguments would be assigned ratings for ethos, the credibility of the source; pathos, the emotional content of the message; and logos, the logical content of the message. This creates a system that parallels the mathematical conflict resolution models typically associated with guns and targets in games, but replaces muzzle flashes with speech acts and ballistics models with persuasion theory. If written carefully, conflicts structured in this manner can play out differently each time a game is played. Furthermore, a system such as this allows for the same work on the part of writers to be used repeatedly in very different scenarios.

The second obstacle mentioned above concerned the choreographic workload. Adding in each hand movement and head nod is both tedious and hopefully unnecessary. What's interesting to observe, looking at the node counts, is that it was really only about twice as many behaviors (gesture commands) to fill in compared to the lines of dialog themselves, but it took substantially longer than twice the amount of time to get done. While the dialog was laid out in sufficient detail that getting it into the editor was only data entry (2 weeks), for gestures, we had to think about what was appropriate when and how to adapt stage instructions that weren't quite feasible. Getting the gestures right really requires working with the game itself to make sure things look like you think they will. An interesting question is whether there might be a way to short circuit this type of effort, particularly as the digital cast is reused from game to game, and as we gain more experience with types of gesture sequences that go with high level behaviors and conversations. Given the conflict system metaphor mentioned in the last paragraph, it seems highly likely that we could coordinate certain gestures with how the character's favored position is faring. We might also look at high level markup of the dialog, such as "Jack is concerned as he is saying this", and the system matches up a list of "concerned" gestures to find an animation to play. Certainly there are many such ideas that would be worth looking into.

Goal 2: Insulate Authors from Engine – Initially, we thought the scene-batch mode would be a useful approach for the HSG team, and that we would benefit from insulating the team from the engine. Indeed, the AESOP generator is relatively independent of any given commercial implementation, and that did prove to be a benefit to our authors. One can safely author the game in the FSM trees and assume that the XML interface will convert the results into the syntax and instructions needed by the respective game engine. However, the engine wrapping side of our AESOP effort was not a small activity. It required about .5 man-year of the 3rd author's time to wrap the Director engine, and another .3 man-year of a separate programmer's time to wrap Unreal Tournament. Very little was reused between those two efforts. A second issue is that the scene-batch mode of authoring is one of the obstacles mentioned above under the choreographic load. That is, at present, the authors must insert a gesture command (behavior) into the FSM tree and then play the game to see how it looks. To improve this batch process, at present we are eliminating the distinctions between the Engine and the various Editors. The ideal we are currently gravitating towards is that content developers and writers can directly manipulate sets and characters within the engine and edit positioning, gestures, dialogs, and player choices in the context of each scene, beat, and dialog string. While there are a number of unsolved obstacles to doing this while preserving engine independence, this interpretive mode should be more gratifying to the developers, making the process less of a chore and more like a direct beat-manipulation interface where they can observe a portion they don't like or haven't finished, backup, edit it, and replay the beat until they get it how they want it. However, there will still be the need for popup windows that are the batch mode viewers, since that supports bigger picture viewing/manipulating of what is being authored for a scene, act, or story.

Goal 3: Reusable Cast – Another sizable challenge is the need for highly composable systems that allow interactive dramas and scenarios to be generated on demand and just-in-time for the purpose of story sharing. This is the "Holodeck" dream, which begs a flotilla of research and development priorities, only some of which have been addressed in this article. We realize that we have only just have begun to move down this path with our current cast of ASEOP characters, behaviors, and lessons learned.

ACKNOWLEDGEMENTS

We gratefully acknowledge the National Heart Attack Alert Program, the National Library of Medicine PO # 467-MZ-902060, and the Beck Fund. Also appreciated is the help of the HSG team (Drs. Holmes, Kimmel, Green, & Holmes) and numerous student helpers.

REFERENCES

[1] Campbell, J., *The Hero With a Thousand Faces*, Princeton: Bollingen Series/Princeton University Press 1973.

[2] Decker, D., "Anatomy of a Screenplay", *First Annual Theory of Story Conference (Storycon)*, Palm Springs: CA, Sept.2002. www.anatomyofascreenplay.com/book.htm

[3] Green, GS, Bavelier, D, "Action Video Game Modifies Visual Selective Attention," *Nature*, v.423, pp. 534-7, May 03.

[4] Green, M, Brock, TC. "The Role of Transportation in the Persuasiveness of Public Narratives," *J. of Personality and Social Psychology*, 79(5), May 2000, pp 701-721.

[5] Hayes-Roth, B, and Rousseau, D., "Improvisational Synthetic Actors with Flexible Personalities." Stanford Knowledge Systems Laboratory Report KSL-97-10, 1997.

[6] Lasseter, J., "Principles of Animation Applied to Computer Animation," Computer Graphics, 21(4), July 1987, pp. 35-44.

[7] LeBlanc, M., "Game Design Workshop," *Game Developers Conference*, San Jose, CA: March 2002.

[8] Marsella, S., Johnson, W.L., and LaBore, C. "Interactive Pedagogical Drama." *Proceedings of the 4th International Conference on Autonomous Agents*, 2000, pp 301-308.

[9] Mateas, Michael. *An Oz-Centric Review of Interactive Drama and Believable Agents*. Technical Report CMU-CS-97-156, CMU, Pittsburgh, PA. June 1997.

[10] Petty, R. E., & Cacioppo, J. T.. *Communication and persuasion,* New York: Springer-Verlag, 1986.

[11] Propp, V, *Morphology of the Folktale*, Austin: University Texas Press, 1968.

[12] Zyda, M., Hiles, J., Mayberry, A., et al., "The MOVES Institute's Army Game Project" *IEEE Computer Graphics and Applications,* January/February 2003.

[13] Toth J, Van Lent, M, Silverman, B., et al., "Leveraging gaming in DOD modeling and simulation", *12th Conf on Behavior Representation in Modeling & Sim* ,SISO, May 2003.

[14] Silverman, BG, Holmes, J, Kimmel, S, et al., "Modeling Emotion and Behavior in Animated Personas to Facilitate Human Behavior Change," *INFORMS' Jnl of HealthCare Management Science*, 4(3), Sept. 2001, 213-228.

[15] Silverman, BG, Johns, M, Weaver, R, et al., "Using Human Behavior Models to Improve the Realism of Synthetic Agents," *Cognitive Science Quarterly,* 2(3/4), Fall'02, 273-301.

[16] Swartout, W., et al. "Toward the Holodeck: Integrating Graphics, Sound, Character and Story." *Proc. Fifth International Conf. Autonomous Agents*, May 2001, pp 409-416.

[17] Silverman, BG, Johns, M, Weaver, R, "Satisfying the Perceived Need for Free-Play in Pedagogically Oriented Interactive Dramas," Conference on Animated and Social Agents Proceedings., New York: IEEE Press, May 2003.