November 2003

Authoring Edutainment Stories for Online Players (AESOP): Introducing Gameplay into Interactive Dramas

Barry G. Silverman
University of Pennsylvania, basil@seas.upenn.edu

Michael Johns
University of Pennsylvania

Ransom Weaver
University of Pennsylvania

Joshua Mosley
University of Pennsylvania

Recommended Citation

Authoring Edutainment Stories for Online Players (AESOP): Introducing Gameplay into Interactive Dramas

Abstract

The video gaming industry has experienced extraordinary technological growth in the recent past, causing a boom in both 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. Section 1 postulates 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 2 and 3 present progress on simple versions of those tools and a case study that made use of the resulting toolset to create an interactive drama.

Keywords

Comments

Authoring Edutainment Stories for Online Players (AESOP): Introducing Gameplay into 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 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. Section 1 postulates 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 2 and 3 present progress on simple versions of those tools and a case study that made use of the resulting toolset to create an interactive drama.

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 (overeating, 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 gameplay 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 gameplay, 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 gameplay 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 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.

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, 5] as well as how they are used in franchise games, comics, and serials. As shown below, we include characters of different ages, genders, and backgrounds/ethnicities, and in the roles of hero, sidekick, allies, opponents, tricksters, lovers, and so on. Our hopes for reuse by many nontechnical authors has driven us to keep most things simple (art, animations, behaviors, etc.).

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 this simple 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. Elsewhere we explore migrating to more complex FSM models and to agents that are in effect infinite state machines as in [7].

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.

2 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. Its 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 1 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 such groups as they attempted
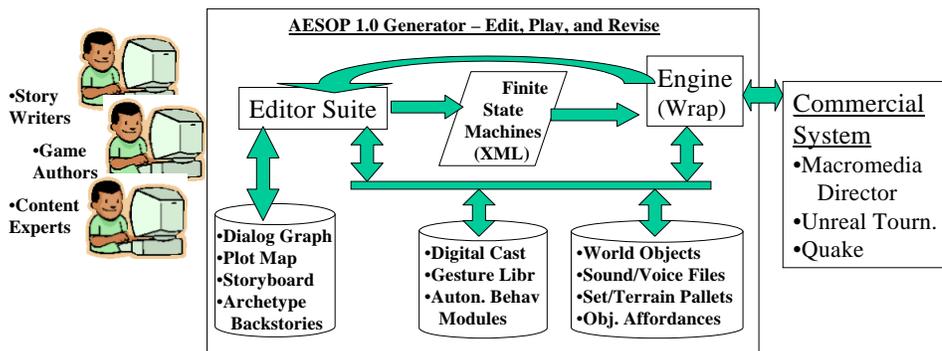

Fig. 1. Architecture of the AESOP Generator to Insulate Commercial Engines and Help Authors Create Interactive Dramas

to create an edutainment system, elicit their design protocols and intermediate game representations, and 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 [20] and the Heart Sense Game (HSG) role playing drama [6]. The application

connects AESOP to an engine written in Director. Another 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.

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 six faculty investigators, two graduate student researchers, at times up to nine undergraduate digital media design and systems engineering students (helping with art, animation, sound, voiceovers, etc.), and one junior and one senior screenwriter (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 1 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 2.1). 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 2.2). 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 [6, 7], while our other applications 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 [7, 8] among others, and we continue to work on the challenges of integrating author- vs. agent-driven story elements.
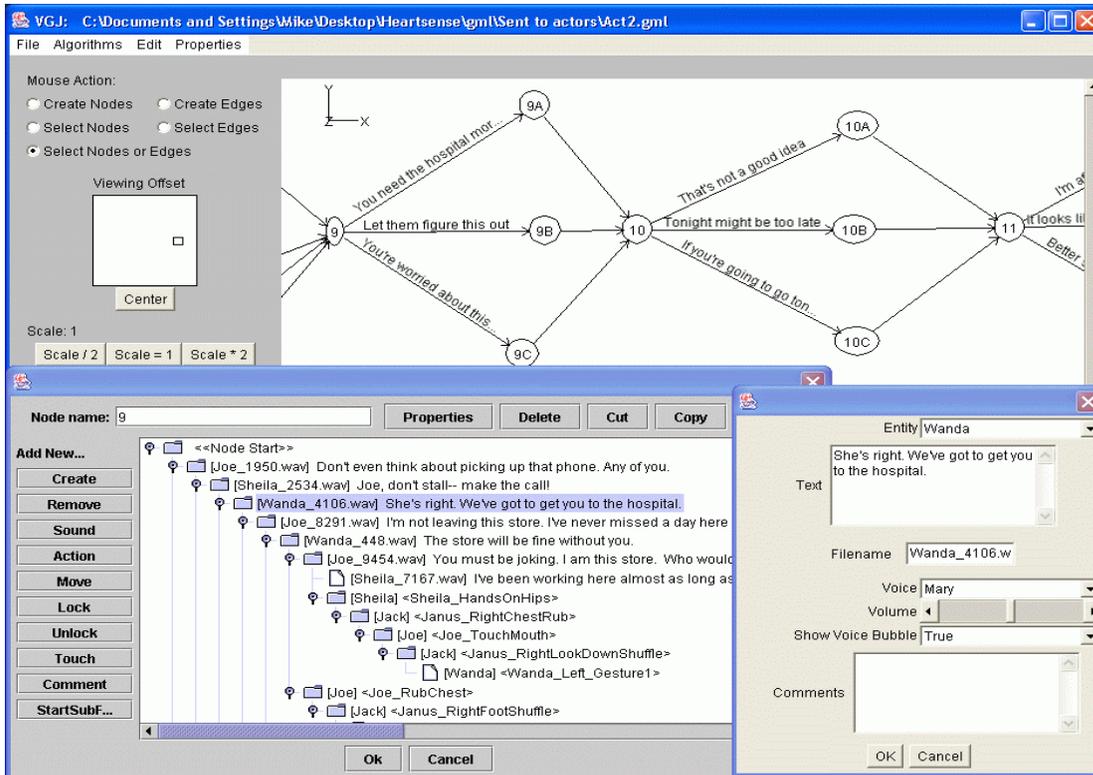
Fig. 2. Select Screens of the GraphEdit Tool for Branching Dialogs and for Adding Choreographic and MultiMedia Instructions to FSMs

2.1 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 (see Figure 2). 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.

2.2 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. Some of the reusable cast members are shown below.

In addition, they created a Director-based Macro Animation Editor, shown partly in Figure 3, 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 eight-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 be integrated with this Macro editor.

2.3 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).

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.

I/O FILENAME:		0	BODY_TURN_FRONT
			BODY_TURN_LEFT
			BODY_TURN_RIGHT
			FACE_NEUTRAL
			FACE_CONTENT
		10	FACE_HAPPY
			FACE_SAD
		25	FACE_ANGRY
			FACE_SURPRISED
			HEAD_TILT_LEFT
			HEAD_TILT_LEFT_RETURN
			HEAD_TILT_RIGHT
			HEAD_TILT_RIGHT_RETURN
			HEAD_TURN_LEFT
717px			HEAD_TURN_LEFT_RETURN
			HEAD_TURN_RIGHT
			HEAD_TURN_RIGHT_RETURN
			EYE_LEFT_OPEN
			EYE_LEFT_CLOSE
BEGIN ANIMATION SEQUENCE			EYE_RIGHT_OPEN
PAUSE ANIMATION SEQUENCE			EYE_RIGHT_CLOSE
STOP ANIMATION SEQUENCE			HAND_LEFT_EXTEND_LOW_LEFT
OUTPUT ANIMATION SEQUENCE			HAND_LEFT_EXTEND_LOW_LEFT_RETURN
			HAND_LEFT_EXTEND_MID_LEFT
LOAD ANIMATION SEQUENCE			HAND_LEFT_EXTEND_MID_LEFT_RETURN
		30	HAND_LEFT_EXTEND_HIGH_LEFT
		40	HAND_LEFT_EXTEND_HIGH_LEFT_RETURN
ENDFRAME NUMBER:			HAND_RIGHT_EXTEND_LOW_RIGHT
CUELIST: (cue:frame,cue:frame)			HAND_RIGHT_EXTEND_LOW_RIGHT_RETURN
FRAMES PER SECOND:			HAND_RIGHT_EXTEND_MID_RIGHT
INITIAL POSITION:			

Fig. 3. Overview of the Flash Macro Editor for Building Gesture Procedures and Movies

3 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 its appeal cannot be summarized in a few 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 writers 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 (one 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, it's 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.

The result of a negotiation between writers and trainers often omits the concept of gameplay since neither are trained in this aesthetic. In this case study, the same thing happened. While the first version of the drama was interactive, it was linear and the player could only choose which of about three possible phrasings to reply with for each situation. The player had no sense of being in control, of being able to alter the outcomes, or of thinking there was a reason to play again. As already mentioned, the lead author started the project with a paper-based version. This involved five notecards, one each for Acts 1 and 3, and three more for the various quests or scenes of Act 2. The front of each notecard involved options for playing the hero as a "good guy" who saves all heart victims, rescues the

kidnappee, helps and is accepted by the townfolk, and then as a result of all this wins the romantic affection of the leading lady and a job as the apprentice to the town doctor. The flip side of the cards is for driving your avatar into the anti-hero role – that of ignoring the heart attack victims in the effort to do things needed to win the respect of and a job in the organization of the antagonists. The player gains a major role in the antagonists' organization, but at the expense that the townfolk disdain you, the romantic interest shuns you, and the kidnappee believes you abetted his abductors. Conflict involves true tradeoffs – something must be sacrificed for gains in another dimension. This is entertaining and it gives the player real choices to make in the drama, not just phrasings of verbiage. With this as background the authors of this paper worked with the writers to try and get them to shift their approach to a non-linear script writing effort.

In the end we found three strong storylines, and got the writers and training content specialists to go along with them so the players could have a sense of 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, the lead author tasked 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 one 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 six person-months equivalent across all the faculty co-investigators (content developers, story critics, and play testers).

4 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, conflict, 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 needs of the target audience. While a game like Heart Sense is inherently dialog-oriented with its focus on persuasion and interpersonal relationships, one would also want the generator to support a wide array of genres. 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. To date, we have tried explicitly to keep the cast and generator simple, and in turn, to limit the features that authors need to conquer. The field needs to move beyond this, to step forward on this generator front, and to rise to the challenge if we are to achieve the goal of many games to help people to learn to cope with the array of health and personal issues that confront them.

References

1. Campbell, J.: The Hero With a Thousand Faces. Bollingen Series/Princeton University Press, Princeton (1973)

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

3. Green, G.S., Bavelier, D.: Action Video Game Modifies Visual Selective Attention. Nature. **423** (2003) 534-537

4. Green, M., Brock, T.C.: The Role of Transportation in the Persuasiveness of Public Narratives. J. of Personality and Social Psychology. **79** (2000) 701-721

5. Propp, V.: Morphology of the Folktale. University Texas Press, Austin (1968)

6. Silverman, B.G., 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** (2001) 213-228

7. Silverman, B.G., Johns, M., Weaver, R., et al.: Using Human Behavior Models to Improve the Realism of Synthetic Agents. Cognitive Science Quarterly. **2** (2002) 273-301

8. Silverman, B.G., Johns M., Weaver R. Satisfying the Perceived Need for Free Play in Pedagogically Oriented Interactive Dramas, Conference on Animated and Social Agents Proceedings. IEEE Press, New York (2003)