Search-Based Drama Management

Ari Lamstein **Michael Mateas**

The Georgia Institute of Technology
College of Computing, Technology Square Research Building
85 5th Street, NW
Atlanta GA 30332-0760
lamstein@cc.gatech.edu michaelm@cc.gatech.edu

Abstract

A major obstacle to creating compelling story-based video games is shaping the player's experience into a narrative structure without impinging on the player's sense of agency (interactive freedom). Story-based video games (e.g. adventure games, many RPGs) tend to prescript linear or mildly branching stories, alternating periods of relatively open-ended and story-free interaction with non-interactive story events (cut-scenes) or discrete story choices. Simulation world games (e.g. *Grand Theft Auto*, some RPGs) present the player with many possible courses of action at every moment, at the expense of global story structure. We propose a system, a Search-Based Drama Manager (SBDM), which can help guide the player's experience in more open-ended story worlds along narratively pleasing story arcs. SBDM rests on two assumptions: that an evaluation function can encode an author's aesthetic for a story, and that search can be used to effectively guide a user's experience in a story world. We present preliminary work on applying SBDM to the Interactive Fiction piece Anchorhead.

Introduction

A major obstacle to creating compelling story-based video games is shaping the player's experience into a narrative structure without impinging on the player's sense of agency (interactive freedom). Story-based video games (e.g. adventure games, many RPGs) tend to prescript linear or mildly branching stories, alternating periods of relatively open-ended and story-free interaction with non-interactive story events (cut-scenes) or discrete story choices. Simulation world games (e.g. *Grand Theft Auto*, some RPGs) present the player with many possible courses of action at every moment, at the expense of global story structure. We propose a system, a Search-Based Drama Manager (SBDM), which can help guide the player's experience in more open-ended story worlds along narratively pleasing story arcs. SBDM rests on two assumptions: that an evaluation function can encode an author's aesthetic for a story, and that search can be used to effectively guide a user's experience in a story world.

In SBDM, a player's concrete experience in the world is captured by a sequence of *Player Moves*, abstract plot points that a player's activity can cause to happen. A single Player Move may encapsulate 5 or 10 minutes of concrete player activity in the world - moving around, picking up objects, interacting with characters and so forth. When the concrete activity "adds up" to a story significant event, then a Player Move is recognized. A SBDM has a set of *System Moves* available that can materially alter the world (e.g. move objects around, change goals in characters' heads, etc.) in such a way as to encourage or obviate a Player Move. System Moves give the SBDM a way to warp the world around the player so as to make certain Player Moves more or less likely. Besides the System Moves, the author also provides the SBDM with a story-specific evaluation function that, given a complete sequence of Player and System Moves, returns a number indicating the "goodness" of the story. Whenever the drama manager recognizes a Player Move (plot point) occurring in the world, it projects all possible future histories of Player and System moves, evaluates the resulting total histories with the evaluation function, and backs these evaluations up the search tree (in a manner similar to game-tree search) to decide which system move to make next that is most likely to cause a good total story to happen.

We present preliminary work on applying SBDM to the Interactive Fiction piece Anchorhead.

Related Work

The idea of Search-Based Drama Management was first proposed by Bates [Bates 1992] and developed in Weyhrauch's dissertation work [Weyhrauch 1997]. Weyhrauch defined a story evaluation function for a simplified version of the Infocom interactive fiction *Deadline*, and developed a game-tree search algorithm that employed the evaluation function. However, his drama manager was never hooked up to a story world. His story search was purely abstract, with a simulated user who randomly made coarse-grained Player Moves, with the

probability modulated by the previous System Move. He also only implemented an evaluation function for a single story world, bringing the generality of the results into question. His work validated the idea that an author-specific story aesthetic could be captured in an evaluation function, but left open the question of whether an SBDM could effectively guide the player's experience within a concrete story world. This work can be considered the successor to his work in that:

- it will be the first full implementation of an SBDM (connection to a concrete story world).
- it will be the first use of an SBDM for a story outside that presented in [Weyhrauch 1997].
- we will evaluate the effectiveness of SBDM with human players in a concrete story world.
- we will explore search techniques to allow this approach to operate in real time.

Script-and-demon story systems combine the use of a script to specify linear or branching sequences of events with demons that are associated with events. The demons won't let an event happen until certain preconditions on the state of the world have been satisfied. The plot is generally represented in a graph or script form, where the author specifies the demons associated with each story scene that decide whether the scene has been accomplished. Examples include plot graphs [Kelso, Weyhrauch & Bates 1993], Interval Scripts [Pinhanez 1997], Galyean's Dogmatix architecture [Galyean 1995], and the story graphs of the Mission Rehearsal Exercise Project [Swartout et. al. 2001]. Unlike SBDM, these approaches only make use of local interaction and don't take account of global story structure. Further, these approaches incur a large authorial cost; the author must pre-specify all the paths through the story.

Rule-based approaches use rules to describe the story structure and/or character actions as rules that trigger on the story state and internal character traits. The Defacto story environment uses rules to describe norms, social actions, personal goals and relations [Sgouros 1999]. The Erasmatron uses rules to describe the relationships between character traits, actions, and the state of the world [Crawford 2003]. Both systems assume discrete player choices; player's choose actions from menus at distinct interaction points. SBDM, on the contrary, by abstracting from the concrete, potentially real-time, player activity, is able to combine a concern for global story structure with continuous and more open-ended interactions.

The Mimesis architecture [Young 2001] constructs story plans for real-time virtual worlds, currently the Unreal Tournament world. The story plans generated by the planner are annotated with a rich causal structure. The system monitors for player actions that might threaten causal links in the current story plan, either re-planning or preventing player action if a threat is detected.

Mateas and Stern developed a Beat-Based Drama Manager for the Interactive Drama *Façade* [Mateas & Stern 2003; Mateas 2002]. Beats are the smallest unit of dramatic value change in drama, where values are character and story attributes such as love, trust, tension, etc. A beat-based drama manager is appropriate for tight story structures, where ideally all the activity in the story world contributes to the story. SBDM is more appropriate for large simulated worlds in which the author would like the experience to have large-scale story structure, but still allow for more open-ended exploration and interaction in the local structure.

The Story

For our initial experiments in hooking an SBDM to a concrete world, we've chosen to work with a text-based interactive fiction (IF). There is a large and vibrant community still writing new IF work, some of it of very high quality. For many IFs, one can obtain the source (often written in Inform) and can thus add support for recognizing Player Moves and executing System Moves. And the discrete time model of such worlds can mask any latencies initially associated with search (one of the goals of this work is to develop search approaches that can work in real time). The next step after applying SBDM to an IF will be to apply it to a mod-able game system such as *Never Winter Nights*, in order to explore issues arising from real-time player interaction.

Many IFs (as well as graphical adventures and RPGs) are organized either as a puzzle-based design in which the player overcomes a collection of obstacles (e.g. puzzles, monsters) in no particular order, or as a story-based design in which the player moves through a linear sequence of obstacles, with no deviation from the sequence possible. Neither design is appropriate for the application of SBDM. SBDM is most appropriate for worlds in which their are many possible sequences through the world, a few of these sequences have better story structure from the rest, but the player is not forced by the causal structure of the game down one of these sequences, and is thus free to deviate from them. The IF we've selected with these properties is *Anchorhead*.

Anchorhead

Anchorhead is a Lovecraft-style horror story written by Michael S. Gentry. It is freely available from Baf's Guide to the Interactive Fiction Archive [Muckenhoupt]. The game was nominated for 6 Xyzzy awards, the annual awards for IF.

The story features a female protagonist who has recently moved to the town of Anchorhead with her husband, Michael. The couple moved there because a previously unknown relative of Michael's, Edward Verlac, passed away and left his mansion to Michael. The story unfolds as the player uncovers the disturbing history of the town, and the role which the Verlac's have played in that history.

The bulk of the game is split into three days, which are further split into day and evening. Time only passes after

the user completes a fixed set of events. There is also a final night and a brief epilogue.

The game itself is very large, with well over 100 Player Moves. Our interest is a proof-of-concept of the efficacy of SBDM, and we have selected only a subset of Day 2 of the game to Drama Manage. By the beginning of Day 2 the Player has seen most of the map and learned that Edward and his family did not die naturally: Edward murdered them and then killed himself.

The rest of the paper describes an idea run through our game, the evaluation function, the search algorithm, and provides conclusions.

An Ideal Run

We use the word "run" to describe a Player's experience in our game. Each run is like running an experiment, where the initial conditions (the game) are fixed, but the outcome (the Player's experience) is uncertain. For the sake of brevity, we discuss only a subset of our game in this paper. Here we describe a run that we, as authors, feel has a nice story structure. We would want our evaluation function to give this run a high score.

The Player wakes up and begins exploring the mansion. She is interested to learn more about the Verlacs, Anchorhead and the Mansion: Day 1 hints that each of these has a dark secret. She finds a series of old newspaper clippings which provide details about Anchorhead and the Verlacs. Anchorhead has suffered from a large number of missing children. Additionally, the Verlacs have had several disputes with their neighbors over the years. It is not clear if there is any relation between these two, or why the Verlacs saved newspaper clippings detailing kidnappings. Seeking to learn more, the Player begins exploring outside the mansion. She meets a cryptic old man who complains about being thirsty. Thinking that he might be able to provide clues about the above themes, she tries to engage him in coversation. However, he refuses to speak to her. This man serves as a mystery: What does he know, and how can the Player get him to speak? She leaves and goes to the library.

There she borrows the same book that Michael was reading yesterday: *A Historical Overview of Superstitions in the Miskaton Valley Region*. It gives information about the Misquat Indians, who had a reputation of being child-stealers. Could an Indian tribe have been responsible for the kidnappings which the Player just read about? It's not yet clear. The book also has a disturbing section about Croseus Verlac, the first Verlac to arrive in the New World in the 1600s. He apparently slept with his daughters in order to have a "pure" heir. Many of these children died soon after birth, and Croseus himself died the day after fathering his first healthy child. She also finds a note, written by Michael, which mentions wanting to check the courthouse records to verify if this birth/death pattern continued in later generations of Verlacs. Apparently Michael knew more about the history of the Verlacs than he led on.

Seeking to uncover more of the mystery, the Player leaves the library and continues exploring. Upon entering the tavern she finds a flask. She then returns to the town square and notices the courthouse. Remembering Michael's note, she enters and finds the Records department, where she learns that the birth/death pattern started by Croseus has indeed continued. She becomes frightened, and decides to go home and talk to Michael. Before reaching the Mansion, however, she meets the old man again. Realizing that her flask might help him, she gives it to him. They then begin talking.

This ends Day 2, and begins Evening 2.

This run is good because the plot gradually unfolds in a dramatic way. Player Moves meaningfully build on each other, and at each point the Player feels that she has several themes which she can explore.

The Evaluation Function

Player Moves

Player Moves are abstractions of a Player's progress in a story. We represent them as Directed-Acyclic Graphs (DAGs) to show "must-proceed" relationships. Figure 1 shows a DAG of all Player Moves in our game. Player Moves not discussed in this paper appear shaded.

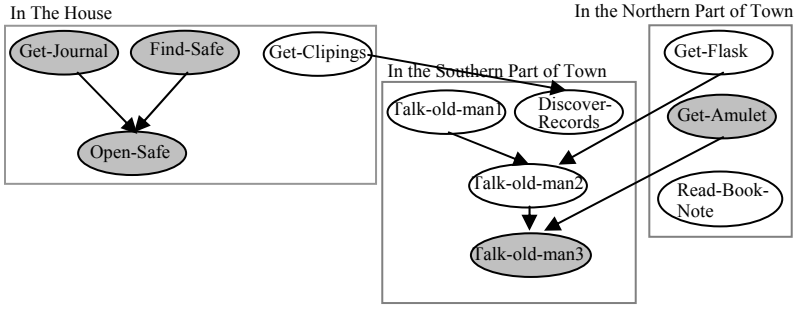

Figure 1. Must-proceed DAG for Player Moves

Player Moves are powerful abstractions, and provide a useful framework for analyzing runs. We argue that the order in which Player Moves occur greatly effects the enjoyment of runs.

Evaluating Runs

The evaluation function takes as input a complete sequence of Player and System Moves. It returns a number which represents the quality of the run. A "complete sequence" is one which contains all Player Moves. The number it returns is the weighted sum of author-defined *features*. The choice of features, including whether or not there exists a canonical set of features, is an active research area for us.

Here we discuss only one feature, *activity flow*, which measures how much a Player has moved back and forth

between various parts of the map. Activity flow was employed by Weyhrauch in his evaluation function [Weyhrauch 1997]. A Player Move has activity flow if it takes place in the same area of the map as the previous Player Move. Table 1 shows the computation of activity flow for our ideal run. Since Get-Flask and Read-Book-Note both take place in Town-North, Get-Flask has activity flow.

A run in which the Player repeatedly criss-crosses the map, such as "Get-Clippings, Talk-Old-Man-1, Get-Flask, Talk-Old-Man-2, Read-Book-Note, Discover-Records", has an activity flow of 0.

Player Move	*Location*	*Activity Flow?*	*Count*
Get-clippings	Mansion	No	0
Talk-old-man1	Town-south	No	0
Read-book-note	Town-north	No	0
Get-flask	Town-north	Yes	1
Discover-records	Town-south	No	1
Talk-old-man2	Town-south	Yes	2

Table 1. Activity flow for ideal run.

System Moves

System Moves are moves the SBDM can make that materially alter the world in order to encourage or obviate a Player Move. We expect to write at least one System Move for each Player Move. (We will consider all Talk-Old-Man Player Moves as the same).

Consider the System Move for encouraging the Get-Flask Player Move. SBDM could simply move the flask to where ever the Player currently is.

We expect the System Move for Discover-Records to be more sophisticated since it requires the Player going to the courthouse. One possibility is having Michael go to the Player's current location and say something which encourages the Player to do this.

We consider the development of a lexicon of effective and non-manipulative System Moves to be an open research question.

Search

The SBDM uses an adversarial search algorithm to decide which, if any, System Move to execute given the recognition of a Player Move. The minimax algorithm of traditional game-tree search is not appropriate for SBDM for several reasons. First, while the drama manager should model itself as knowledgeable (knowing all the rules), the player should be viewed as naive. The player does not know that it is playing a game with the drama manager, and does not want to "beat" it. Indeed, if the evaluation function is correct, then the user "beating" the drama manager would give the user the worst possible experience. Thus, instead of implementing minimax, and assuming that the opponent will make moves which hurt the drama manager, Weyhrauch [Weyhrauch 1997] proposed using avg-max, assuming at leaf nodes that the player will choose any of the possible legal moves. Second, moves don't alternate. The player, as well as the drama manager, can make multiple moves in a row. Third, there is no static evaluation function: the evaluation function is valid only for complete scenarios. Finally, SBDM does not view interactive drama as a win-lose game.

Because the evaluation function requires complete sequences of Player and System moves, search must be full-depth. Such full-depth search will often be intractable in real-time and semi-real-time settings. An advantage of this approach, however, is that once drama management has been cast as a game-tree search problem, many of the techniques already developed in the literature, such as sampling search, static evaluation functions applied to limited depth search, and memoizing search via off-line policy learning (e.g. [Tesauro 1995]) become applicable to this problem.

Conclusions

SBDM is an approach for bringing large-scale story structure to game worlds while preserving the open-endedness of the local interaction. SBDM will enable new game experiences by allowing authors to include story-structure in games without having to force the player through the story - if the player doesn't end up experiencing one of the preferred sequences, nothing breaks, but the SBDM helps to make the preferred sequences more likely.

References

Bates, J. 1992. Virtual Reality, Art, and Entertainment. *Presence: The Journal of Teleoperators and Virtual Environments* Vol. 2 No. 1: 133-138.

Crawford, C. 2003. Assumptions underlying the Erasmatron storytelling system. In M. Mateas and P. Sengers (Eds.), *Narrative Intelligence*. Amsterdam: John Benjamins.

Galyean, T. 1995. *Narrative Guidance of Interactivity*. Ph.D. Dissertation, MIT Media Lab, MIT.

Kelso, M., Weyhrauch, P., Bates, J. 1993. Dramatic Presence. *Presence: The Journal of Teleoperators and Virtual Environments*, Vol. 2 No. 1, MIT Press.

Mateas, M. 2002. *Interactive Drama, Art and Artificial Intelligence*. Ph.D. dissertation. Tech Report CMU-CS-02-206, Carnegie Mellon University.

Mateas, M and Stern, A. 2003. Integrating plot, character and natural language processing in the interactive drama Façade, In *1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment* (TIDSE '03), Darmstadt, Germany , March 24 – 26, 2003 .

Muckenhoupt, C. Baf's Guide to the Interactive Fiction Archive. http://www.wurb.com/if/.

Pinhanez, C. 1997. Interval Scripts: a Design Paradigm for Story-Based Interactive Systems. In *Proceedings of CHI97*. Atlanta, GA, pp. 287-294.

Sgouros, N. M. 1999. Dynamic generation, management and resolution of interactive plots. *Artificial Intelligence* 107 (1999): 29-62.

Swartout, W., Hill, R., Gratch, J., Johnson, W. L., Kryakakis, C., LaBore, C., Lindheim, R., Marsella, S., Miraglia, D., Moore, B., Morie, J., Rickel, J., Thiebaux, M., Tuch, L., Whitney, R., Douglas, J. 2001. Toward the Holodeck: Integrating Graphics, Sound, Character, and Story. In *Proceedings of the 5th International Conference on Autonomous Agents*.

Tesauro, G. J. 1995. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38:58-68.

Weyhrauch, P. 1997. *Guiding Interactive Drama*. Ph.D. Dissertation, Tech report CMU-CS-97-109, Carnegie Mellon University.

Young, R. M. 2001. An Overview of the Mimesis Architecture: Integrating Intelligent Narrative Control into an Existing Gaming Environment. In *The Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*.