Integrating Drama Management into an Adventure Game

Anne Sullivan, Sherol Chen, and **Michael Mateas**

Expressive Intelligence Studio
University of California Santa Cruz
Santa Cruz, California
{anne, sherol, michaelm}@soe.ucsc.edu

Abstract

Often, video game designers must choose between creating a linear experience, and designing an open world with many different story lines that fail to form a tightly crafted narrative arc. A drama manager (DM) can provide a solution to this dilemma. A DM monitors an interactive experience, such as a computer game, and intervenes to shape the global experience so that it satisfies the author's expressive goals without decreasing a player's interactive agency. In this demo we present the first integration of declarative optimization-based drama management (DODM) into an adventure-style dungeon game called EMPath.

Introduction

When creating a video game, designers are faced with the decision of how to integrate a compelling and cohesive story line for the player to experience. Currently, most commercial games tend towards one of two strategies. The first solution is creating a fairly linear experience which allows great control over the player's choices which ensures a cohesive story. The second strategy is creating a very open world that allows the player many options but allows for less authorial control and therefore, less cohesion within the experience.

A drama manager (DM) is an alternative solution to this dilemma. A DM makes it possible to balance authorial goals without decreasing the player's agency. It accomplishes this by monitoring the game play and then intervening to shape the global experience. Within existing game genres, drama management can perform tasks such as non-linear mission, encounter, and player discovery sequencing. Drama management can support a level of non-linearity in existing game genres that would be difficult to impossible to achieve using the traditional approach of scripting and local triggers. Additionally, drama management opens up new game design possibilities that would traditionally be complicated or impossible to conceive of without the capabilities of a DM. In this paper we report on the application of declarative optimization-based drama management (DODM) to a *Zelda*-style adventure game called EMPath. This is the first integration of DODM into a playable game.

The DODM Framework

To configure DODM for a specific world, the author specifies *plot points*, *DM actions*, and an *evaluation function*.

Plot points are important events that can occur in an experience. Different sequences of plot points define different player trajectories through games or story worlds. Examples of plot points include a player gaining story information or acquiring an important object. The plot points are annotated with ordering constraints that capture the physical limitations of the world, such as events in a locked room not being possible until the player gets the key. plot points are also annotated with information such as where the plot point happens or what subplot it is part of. The exact set of annotations is flexible and depends on the story.

DM actions are actions the DM can take to intervene in the unfolding experience. Actions can *cause* specific plot points to happen, provide *hints* that make it more likely a plot point will happen, *deny* a plot point so it cannot happen, or *un-deny* a previously denied plot point.

The evaluation function, given a total sequence of plot points that occurred in the world, returns a "goodness" evaluation for that sequence. This evaluation is a specific, author-specified function that captures story or experience goodness for a specific world. While an author can create custom story features, the DODM framework provides a set of features that are commonly useful in defining evaluation functions.

As examples, *Motivation* measures whether events that happen in the world are motivated by previous events. For instance, in EMPath, the player can learn from a note that they must acquire a candle before leaving the dungeon. If they encounter this note before finding a boss monster in a room with candles, then the encounter with the boss is motivated. Similarly, *Thought Flow* measures the degree to which plot points associated with the same topic appear together. In EMPath, this feature prefers stories in which plot points associated with each of the quests are grouped together without interleaving. *Manipulation* measures how coercive the Drama Manager is in a given story sequence. Manipulation costs are associated with each DM action rather than being associated with plot points. Manipulation counter-balances other features, forcing the DM to take into account manipulation costs when optimizing other features. The author associates plot points or DM actions with specific evaluation features.

When DODM is connected to a concrete game world, the world informs the DM when the player has caused a plot point to happen. The DM then decides whether to take any actions, and tells the world to carry out that action. In EMPath, for example, the player may be moving between rooms and fighting non-boss enemies. When the player

finds a note describing a nearby prisoner, the game tells the DM that a plot point has happened. The DM might then choose the "deny candle location information" action, telling the game world to remove a specific note.

Given this model, the DM's job is to choose actions (or no action at all) after the occurrence every plot point so as to maximize the future goodness of the complete story. For EMPath, we perform this optimization using game tree search in the space of plot points and DM actions, using expectimax to backup story evaluations from complete sequences. In order to perform this look-ahead search, DODM requires a player model to predict future player action at the plot point level. For more information on the DODM framework see (Nelson and Mateas 2005; Nelson et. al. 2006; Nelson et. al. 2006).

EMPath

EMPath (Experience Management Path) is a classic *Zelda*-like adventure game that we developed specifically to test DODM in a traditional game genre. EMPath was designed to be small, so that it can be completed quickly and players can experience a clear sense of progression and completion from beginning to end. It also was designed to be playable without the DM, yet amenable to drama management.

Game Description

The EMPath world is a 25 room dungeon populated with enemies, fire traps, a prisoner, and two bosses. The goal is to reach the stairs and escape from the dungeon, however the stairs are too dark to traverse and blocked by rubble; the player must find the special items required to escape.

There are two quest lines in the game. The first involves finding a candle so that the player can see their way up the stairs. There are notes that describe the location and use of the candle. The candle is in the possession of the monkey king, whom the player must kill to acquire the candle.

The second quest line revolves around a prisoner, who possesses a magic talisman that can clear the rubble blocking the stairs. A note in the game world informs the player of the existence of a prisoner imprisoned in a jail cell in the dungeon. When the player finds the prisoner, they are told that the large guard next door (boss) has the key. Once killed, the guard will drop the key to the cell. When the player frees the prisoner, they receive the magic talisman.

Integrating DODM into a Game

Prior DODM research has focused on abstract game models defined purely in terms of plot points and DM actions. In connecting DODM to a concrete game world for the first time, we had to modify DODM to account for aspects of the player's experience influenced by the physical layout of the world. This involved the creation of a new evaluation feature and a modification to the player model.

Prior to EMPath, all the DODM evaluation features we developed had no dependencies on physical details of the game world. As we defined the evaluation function for EMPath; however, we realized that none of the features in our toolbox adequately capture a notion of *story density*. Story density is a measure of how many plot points happen per unit time. As authors, we felt that for EMPath long periods of wandering between plot points should get a lower evaluation score, as should plot points happening too close together. For a room-based dungeon crawl like EMPath, we captured "time" in terms of the number of room transitions between plot points.

To accommodate the new story density feature, the player model must be modified to predict the number of room transitions between plot points. To support this, the DM maintains a simplified model of the game world, containing the locations where plot points can happen. The player model estimates the number of room transitions between projected plot points as 1.5 times the Manhattan distance between plot points. However, DM actions can influence the location at which plot points might happen. To project possible futures, the player model must estimate where the plot points affected by DM actions will happen. To illustrate how we handle this, consider the case of two types of hints used in EMPath: hints that drop notes in the next room, and hints that cause the next randomly encountered enemy killed to drop a note ("enemy drop" hints). They are hints rather than causers because, even if the DM drops a note in front of the player, the DM can not guarantee that the player will pick it up.

For the "next room" hint, the player model predicts a randomly chosen adjacent room as the location where the note will be found. For the "enemy drop" hint, we use a probabilistic model that computes the likelihood of the player engaging and killing an enemy, and the estimated "wandering distance" between the player's current location and the closest plot point, to compute the probability that the enemy drop hint occurring before any other plot point.

Player Evaluation

We have preformed preliminary player tests with EMPath. While evaluation work is ongoing, we have preliminary results indicating that drama management has a positive effect on the player's experience of the game.

References

Nelson, M.J. and Mateas, M. 2005. Search-based drama management in the interactive fiction Anchorhead. *Proceedings of the First Annual Conference on Artificial Intelligence and Interactive Digital Entertainment*, AAAI Press.

Nelson, M.J. and Roberts, D.L. and Isbell Jr, C.L. and Mateas, M. 2006. Reinforcement learning for declarative optimization-based drama management. *Proceedings of the fifth international joint con-ference on Autonomous agents and multiagent system*: 775-782

Nelson, M.J. and Mateas, M. and Roberts, D.L. and Isbell Jr, C.L. 2006. Declarative Optimization-Based Drama Management in Interactive Fiction. *IEEE Computer Societ*: 32-41.