# Desiderata for Managers of Interactive Experiences: A Survey of Recent Advances in Drama Management

David L. Roberts and Charles L. Isbell

College of Computing
Georgia Institute of Technology
85 5th St. NW
Atlanta, GA 30332
Email: {robertsd,isbell}@cc.gatech.edu

**Abstract:** In recent years, there has been a growing interest in constructing rich interactive entertainment and training experiences. As these experiences have grown in complexity, there has been a corresponding growing need for the development of robust technologies to shape and modify those experiences in reaction to the actions of human participants. One popular mechanism for addressing this need is through the use of a drama manager. A drama manager is a coordinator that tracks narrative progress in the environment and directs the roles and/or responses of objects and agents in order to achieve a specified narrative or training goal. In this paper, we provide a survey of recent advances in drama management technologies for interactive entertainment, and describe a set of desiderata for the qualitative analysis of such systems.

## 1. Introduction

The demand for software systems that support increasingly rich and engaging entertainment and training as reached new levels. While simple computer games of skill remain wildly popular, there is increasing demand for immersive *experiences* that are more akin to stories. These narrative experiences are complex, requiring autonomy for players to influence the way in which the experience unfolds but necessitating some control to preserve coherence and quality. With this increased complexity comes a notable increase in the difficulty of authoring consistent experiences.

In this paper we focus on *interactive drama*, an entertainment experience where the player is an active participant in how a story unfolds. Players exercise autonomy in their interaction by choosing to explore different parts of the environment, engaging other players or non-player characters in some way, and taking specific actions. The environment (*e.g.*, objects in the world, the world itself, or other characters) reacts to the behavior of the player. This makes the experience interactive and player-driven. On the other hand, authors of these experiences design specific situations or plot sequences that they hope will occur during play. Thus there is authorial intent to create a *narrative* quality. It is the combination of these two features that creates interactive drama.

There is a natural tension between player autonomy and designer intent: preserving designer intent necessitates removing player autonomy while ensuring player autonomy makes preservation of designer intent difficult. In the earliest systems, this tension has been addressed through the authoring of an exhaustive set of *local triggers* that provide instructions for the game world and non-player characters (NPCs) to react to the autonomous actions of the player; however, this approach simply does not scale. Recently, the job of mediating this tension has fallen to a more centralized *drama manager* (DM), an omniscient coordinator that directs objects and characters in the game world to influence the plot progression. An omnipotent micromanaging drama manager corresponds to the traditional notion of drama while no drama manager corresponds to a fully autonomous experience. A drama manager that infrequently takes actions to influence—rather than deterministically modify—the experience corresponds to interactive drama.

The idea of using a manager to guide dramatic experiences was first proposed by Laurel [17]. Since then there have been a number of concrete implementations of the idea (see [22] for a somewhat dated survey). In this paper, we will survey a number of systems, focusing on more recent developments and discussing some of their similarities and differences. In addition, we hope extend work in this field by providing a basis upon which to compare these systems. In particular, we describe a number of desiderata we feel are important metrics for the objective qualitative evaluation of these systems and situate each system according to those metrics.

## 2. Desiderata

The subject of how best to evaluate a drama manager is a topic of some debate in the interactive drama community. One concern arises from the need to separate the quality of authoring from the quality of authorial tools. For example, if it is found that players do not rate their experience more highly when a DM is used, it may just be that the author has created a deeply satisfying (or unsatisfying) experience and the DM cannot significantly change the quality of the experience. Alternatively, perhaps a drama manager could improve the quality of the experience if only the tools available to the author allowed her to be more expressive. Another problem arises when we try to separate the quality of the authorial experience from the quality of the player experience. It is not clear who has the highest priority. As we shall see, most systems assume just a model of player behavior and leave it at that.

In addition, there is a choice of analysis between system-building and game analysis/criticism. Generally speaking, system builders are concerned with technical issues related to the process and problems associated with actual implementations of these systems. As such, some of the techniques surveyed in this paper are integrally tied to a particular game system. On the other hand, game analysis is more concerned with looking at the features or affordances of a particular approach to drama management. These techniques tend to be presented independent of a particular game system.

For our purposes, we focus on game analysis. Where it is possible, we have tried to separate the approach from the particular game system. Further, we assume that the author has created a generally pleasing narrative, so that we can evaluate the drama management systems themselves. Note, however, that any analysis remains speculative in that our qualitative analysis characterizes the potential of a drama management system and the affordances it provides to open new avenues for authorship rather than characterizes the degree to which authors can actually exploit those affordances.

Our desiderata are:

- **Speed**: players should not perceive any delay in game action due to decision making by the drama manager.
- **Coordination**: non-player characters in games should coordinate to enhance the experience of the player characters.
- **Replayability**: the game experience should be varied but retain high quality, even during repeated play.
- **Authorial Control**: a drama manager should provide a way for the author to influence the experience of the human player.
- **Player Autonomy**: players should not be so constrained by the drama manager that they cannot pursue their own goals.
- **Ease of Authoring**: the burden of authoring high quality dramatic experiences should not be increased because of the use of a drama manager.
- **Adaptability**: a player's individual characteristics should be exploited to increase the quality of the experience.
- **Soundness**: the drama manger should be amenable to theoretical inquiry, supporting the ability to make verifiable claims.
- **Invisibility**: the drama manager should not appear overly manipulative to the player.
- Measurabilty: the system should provide affordances for measuring author's satisfaction with the authoring process and the set of stories experienced by the player as well as the player's satisfaction with the overall experience.

The desiderata were motivated by three factors: 1) Our work in building systems for managing interactive narrative; 2) The motivations discussed by the authors of the systems we survey in this paper; and 3) Numerous discussions with researchers well versed in game and narrative rhetoric. It is important to note that some of these desiderata are in conflict. For example, *player autonomy* and *authorial control* are well known to be in tension with one another [8,41]. When implementing a particular approach to drama management, a trade off is unavoidable. Of course selecting an approach for any particular case is dependent on what is most appropriate for the particular application. Thus, in general, no one of the desideratum is more important than any other.

Unfortunately, due to space constraints, a complete evaluation of every system using all of the desiderata is not possible. Thus, for each system we survey, we will select two or three of the most applicable desiderata and briefly evaluate that system for just those desiderata.

## 3. Drama Manager Components

To facilitate clear comparisons, we briefly describe components common to all drama management techniques. All drama management approaches are based on: a set of *plot points*; a set of *drama manager actions* that can be taken in the game world; a *model of player responses* to DM actions; and a *model of the author's intent*.

Plot points represent significant game events such as finding a key to a door. Plot points can have precedence constraints to avoid nonsensical situations (*e.g.*, a player entering a locked room without having found the key). A story is thus a valid sequence of plot points. Note that not all plot points need occur to be a valid story.

Drama manager actions provide a way to steer a story toward a "good" sequences of plot points. These actions need not have direct concrete implementations in the game world. For example, a concrete DM action could be removing an object from the game world or causing an NPC to start a conversation. On the other hand, an action could be instructing an NPC to prevent a player character from crossing the street. In this case, the details of how to concretely accomplish this task in the game world are up to the (possibly semi-autonomous) NPCs.[1] Regardless of the implementation, the DM actions are the tools with which the drama manager influences narrative flow.

In order for the DM to reason about action selection, it must have a model of how actions affect the world. In particular, if the DM determines that a player is deviating too far from a desirable plot sequence, it must know which of the many actions available will best guide the player back toward a good sequence. Further, it must know enough to balance between gentle guidance that may not succeed and more heavy-handed actions that will succeed but may be overly apparent. For example, if the author intends for the player to enter a particular building, the DM would not want to take an action to block the entrance, nor would it want to take an action that would clearly be herding the player into the building. Perhaps the DM would create an event that generates sounds from within the building, raising the player's interest in entering.

Finally, all DM systems must have a model of the author's intent. The model must be simple to describe and modify, but expressive enough that the system can use it to choose proper actions.

## 4. Optimization-Based Systems

[1] In this vein, drama managers are similar to agent coordinators. NPCs are (possibly semi-autonomous) agents in a multiagent system communicating with a central coordinator to bring about a high level goal.

The techniques we describe in this section all use an optimization-based idiom for obtaining authorial intent. Specifically, authorial intent is specified in terms of an *evaluation function*. The drama manager selects from its available actions guided by the goal of optimizing this target function. Although originally rooted in traditional AI search techniques, current systems have borrowed heavily from statistical machine learning. This is in distinct contrast to the planning-based systems described later.

### 4.1 Search-Based Drama Management

*Search-Based Drama Management* (SBDM), is attributable to Bates [6] but was studied in greater detail by Weyhrauch [46]. SBDM is based on an abstraction of a game into significant plot events with precedence constraints encoded in a *directed acyclic graph* (DAG). The edges in the DAG do not imply that a particular plot point must occur immediately after its parent in the graph, only that it must not occur before (if it occurs at all). Plot points are also annotated with information about the story such as the location in the story world where the plot point occurs or the dramatic tension that the player is likely to experience. Any sequence of plot points consistent with a topological ordering of the DAG is a valid story.

Game play in this framework proceeds in an alternating fashion with the player triggering plot events and the drama manager taking actions in response. The DM actions in this framework act on a particular plot point. There are a few types of actions: *cause, deny, temp_deny, reenable*, and *hint*. The cause action causes a plot point to occur in the game whereas a deny action prevents a plot point from ever occurring. The temp_deny action suspends a plot point from occurring until a reenable action is applied to it. The hint action is modeled to increase the likelihood that the particular plot point it operates on will occur. Additionally, at each decision point, the DM can choose not to act, allowing the player to be the sole influence on plot progression.

Player responses to DM actions are specified as a transition model between plot events. In this model, a coefficient is associated with each plot point. When a DM action hints at a certain plot point, the hint action has the effect of multiplying the coefficient associated with that plot point by a fixed amount. Then, the probability of the player experiencing a plot point is calculated by normalizing the coefficients associated with all of the plot points that have satisfied precedence constraints.

Lastly, the author supplies an evaluation function defined over a valid sequence of plot points and DM actions. In the literature, this evaluation function is defined as a linear combination of story features. Example features include *location flow, thought flow, manipulativity* and *plot mixing*. The output of this evaluation function is a measure of how good the story is in the eyes of the author—it does not reflect player preference.

Weyhrauch uses *SAS+*, a variant of the expecti-max game-tree search algorithm to optimize the evaluation function. A tree structure is constructed by alternating levels of plot point nodes with DM action nodes. Search alternates maximizing nodes at the plot point levels with expectation nodes at the DM action levels. Two variants of this search are proposed. The first requires exploiting symmetries in the story space to construct a memoization table that enables the author evaluations over complete stories to be propagated up from the leaves of the tree to interior nodes. The second is a fixed depth search (generally short of the full depth of the tree) that uses a set of sampled complete stories as a heuristic estimate of value of the node at which the search terminates.

Lamstein & Mateas proposed revising this technique [16], and Nelson & Mateas further explored it by attempting to reproduce its results [33,34]. In this work, they uncover the difficulty that can arise when authoring a set of actions that will appear consistent with the situation in the game. For example, suppose one of the plot points occurs when an NPC starts a conversation. If the DM takes an action to cause that plot point when the particular NPC is not near the player, then the outcome could ruin the aesthetic of the story. To handle this situation, they add location tags as properties of actions. They were able to reproduce Weyhrauch's results, but found that the technique did not scale well.

Due to the combinatorial complexity of game tree search it is unsurprising that this system does not do well in terms of its **speed**; however, the designers took care to mediate this difficulty by imposing time limits on search and using heuristic evaluation. This system is especially **measurable**. Along with its derivatives described below, this approach to drama management provides a basis for characterizing the success of the drama manager in meeting the goals of the author as expressed by the evaluation function. This has typically been done by calculating the frequency of the different evaluations of the stories that are realized when the DM is used.

### 4.2 Declarative Optimization-Based Drama

Nelson *et al.* continue work on SBDM by introducing *declarative optimization-based drama management* (DODM) [34,35]. In this work, the plot point abstraction, DM actions, player transition model, and author evaluation function are exactly as in SBDM; however, the SAS+ sampling search is replaced with a policy obtained by solving a Markov Decision Process (MDP). MDPs provide a mathematical framework for modeling an online decision making problem when the dynamics of the world are stochastic [14]. An MDP is specified by a set of states, set of actions, a stochastic transition model encoding dynamics, and a reward function. The solution to an MDP is a policy dictating the optimal choice of action in every state that will maximize the long term expected reward. In this formulation of a drama manager, each of the components corresponds to a piece of an MDP specification. The current history of plot points and DM actions define state; the DM actions define a set of actions; the player model defines a probabilistic transition model; and the author's evaluation function defines a reward function. The solution to the MDP represents the optimal choice of action for the DM given any history of plot points and DM actions.

Unfortunately, reinforcement learning is susceptible to a phenomenon common to optimization techniques known as local maxima. Due to the stochastic nature of the game dynamics, it is likely that the policy arrived at will not actually be the optimal policy. Thus, *Self-adversarial Self-cooperative Exploration* (SASCE) was developed specifically to help find solutions to MDPs that best avoid "bad" parts of the state space. The idea behind SASCE is to use the current estimate of the state-value function that defines the MDP policy to select player transitions that are adversarial. In other words, the actual player model is not used in learning the SASCE policy. Instead a "self-adversarial" player model is substituted that forces the DM to learn a policy that optimizes for the worst possible player behavior. Results obtained by simulating game play against the actual player model indicate that this approach helps to reduce the frequency of poorly rated stories while increasing the number of moderately rated stories.

In contrast to SBDM, DODM has an advantage in terms of runtime **speed** because a policy specifying drama manager actions for every situation is computed before game play; however, it does require significant offline computational effort to learn a policy. Like SBDM, it also provides an affordance for **measurability**. Further, reinforcement learning is theoretically well-grounded and sound. Experiments suggest that DODM improves performance; however this appears to come at the cost of **replayability**. The system finds a narrow set of good stores and drives the player towards them.

### 4.3 TTD-MDPs

*Targeted Trajectory Distribution MDPs* (TTD-MDPs) are a variant of MDPs developed specifically to address the issue of replayability [7,9,42].[2] A TTD-MDP is defined similarly to an MDP by: a set of trajectories that represent sequences of MDP states; a set of actions; a stochastic transition model; and a target distribution specifying a desired probably for every complete trajectory. The solution to a TTD-MDP is a stochastic policy providing a *distribution* over actions in every state such that under repeated play the sequence of states will match the target distribution as closely as possible.[3] Any finite-length discrete-time MDP can be converted to a TTD-MDP by simply encoding the history of MDP states into the TTD-MDP trajectories. This results in a TTD-MDP where each trajectory represents a sequence of states in the underlying MDP, optionally including a history of the actions taken.

The specification of authorial intent is a bit trickier in TTD-MDPs. Thus far, there have been two approaches taken: converting the evaluation function and using a set of prototype trajectories.

**Evaluation-based:** Roberts *et al.* present a method for converting the author's evaluation function into a probability distribution over stories [42]. Because the evaluation function is not typically generative, they present an approach that estimates a target distribution. First, a set of stories is sampled uniformly—ignoring stories that evaluate too poorly—and used to construct a "trajectory tree." Probability mass is assigned by normalizing the evaluation scores across all the leaves in the sampled tree. These probabilities are then propagated up the tree to produce a probability for partial stories. Thus, when the DM selects actions according the probabilistic policy described above, it is actually targeting stories in proportion to their evaluation quality.

**Prototype-based:** Cantino, Roberts & Isbell extend TTD-MDPs by introducing an alternative authorial idiom based on a pre-specified set of desirable stories [9]. In this work, they replace the conversion process with a mixture of Gaussians (MOG) model. Specifically, rather than a function that attaches value to a story, the author specifies a set of good prototype stories and defines a distance measure between stories. Each prototype becomes the centroid of a (possibly multivariate) Gaussian distribution. The probability mass that represents the "desirability" of a story is assigned by first determining its distance from each centroid.

This approach is amenable to even more **authorial control**. Specifically, each prototype can be treated differently, assigning unique (potentially non-uniform) mass in the MOG and unique variance. Thus, the authorial question becomes that of providing a small set of desirable stories and indicating a level of desirability. Further, the extent of the Gaussian can be tweaked to emphasize different aspects of stories. In this model, the author can adjust the allowed deviation in any direction by adjusting the values in the covariance matrix associated with each centroid.

TTD-MDPs have proven quite good at addressing **replayability**. Unfortunately, there is potentially a cost in the **ease of authoring**. Inducing distributions using evaluation functions may be no more difficult than defining an evaluation function for other DODM approaches. In fact, providing prototypes may even be easier; however, it is unclear that most authors will find it easy to define game-specific distance measures that capture the nuances of their intent.

## 5. Planning-Based Architectures

Optimization-based approaches are predominantly derived from statistical machine learning methods. In this section, we discuss other approaches that have roots in more traditional AI planning techniques.

### 5.1 Interactive Drama Architecture

Magerko & Laird describe a framework called the *Interactive Drama Architecture* (IDA) [18,19,20,21]. In their system, narrative goals are defined by the author at varying degrees of detail and the job of the drama manager (called the story director) is to ensure that the player's actions do not threaten their realization. For example, suppose the author intends for a particular NPC to provide an object to the player near the end of the story. If the player character meets this particular NPC early in the game and chooses to fire a gun at it, the story director must intervene to prevent the bullet from killing the NPC. IDA uses semi-autonomous SOAR agents [15] that enable the directions from the DM to be made at various levels. Thus, in this case, the DM could instruct an agent to simply "prevent the death" of the NPC and allow the agent to determine how. On the other hand, the DM could provide specific instructions such as "make the pistol jam."

---

[2] The work of van Lent et al. also seeks to address replayability using a two level planning system: a strategic or deliberative level and a tactical or reactive level [45]. Unfortunately, this approach is designed for adversarial games and seems ill-suited to plot-driven open world games where drama managers are typically used.

[3] Closeness is typically an error measure such as KL-divergence.

In either case, a successful outcome preserves the author's goals.

In this system, plot events are labeled with preconditions in the form of logical statements. This approach supports dynamic runtime binding. For example, plot events can be authored with a variable x that appears throughout the story. When the player causes the first plot event that contains x to occur, it is bound to a concrete entity in the game world. This ensures that all subsequent plot events using that variable preserve narrative consistency while minimizing **authorial effort**. This type of runtime adaption is not a feature of the optimization-based systems described above.

Additionally, these logical statements can indicate temporal extent: particular plot events can have a range of discrete times between which they must occur. Thus, if a player is too early or too late in causing a plot event, the DM will recognize this as a threat to preconditions and can intervene. Interestingly, there is no notion of *explicit causality* in IDA. In other words, the DM cannot cause plot events to occur, but can prevent player actions that will preclude plot events from occurring. IDA reasons about *potential threats* using a *predictive player model*. Thus, the game world is a large unstructured space. But, through proactive modification of the game world, the drama manager limits the player to the portion that is consistent with the author's narrative intent: the player has complete autonomy provided they remain within the scope of narrative intent.

IDA's most significant quality is **invisibility**. One side effect of IDA's approach is a potential increase in the player's perception of autonomy. This characteristic is subjective and has not been explicitly measured. Similarly, some aspects of **ease of authoring** also remain unmeasured. It is an open question whether predictive player models can be easily constructed by the non-expert.[4]

### 5.2 Mimesis

Young *et al*. have developed the Mimesis system [10,36,47,48,50,51], a planning system for drama management. A fairly complex architecture, Mimesis is primarily a run-time behavior generator. Mimesis works at multiple levels of abstraction and therefore brings together two representations for action: the procedural representations used by game engines and the declarative representations used by AI planning systems. In contrast to the architectures described earlier, Mimesis does not select the goals to pursue; it develops plans that are implemented at various levels of abstraction in the game to achieve the goals that are selected for it.

In contrast to IDA, Mimesis is reactive. Suppose the player obtains an object that an NPC needs to carry out a plan. If the NPC continues with the existing plan that is dependent on that object, it will fail. To account for this, Mimesis will either repair the NPC's plan through re-planning or temporarily alter the effects of the player's actions to prevent it from obtaining the object. Note, that Mimesis will not attempt to predict that a player will take an action to threaten a plan; however, it will notice that the outcome of an action taken in the world threatens an existing plan.

As mentioned above, Mimesis constructs plans at multiple levels of abstraction. In a functioning system, the request for a plan comes from the game engine, in the form of a set of goals and actions in the story world. The request is handled by the story world planner. This level is implemented using DCOPL, a hierarchical refinement planner. The story plan is then passed back to the game engine as well as a discourse planner [49]. The game engine executes the parts of the story plan that pertain to characters, objects in the world, and the environment in general. On the other hand, the discourse planner constructs a complementary plan to control the music, camera angles, and other auxiliary aspects of the game experience. The combination of the story plan and the discourse plan form a coherent narrative plan that when executed by the execution manager will achieve the game engine's requested goals.

Mimesis is similar in nature to IDA; however, it allows more **player autonomy**. Its approach of reactive repair also allows for greater **adaptability** as well. On the other hand, it lacks **invisibility**. The failure mode of this approach can often result in the kind of intervention that is apparent to the player.

### 5.3 Narrative Mediation

Riedl *et al*. have developed *narrative mediation*, a technique where a story is defined by a *linear plot* progression and by player choices [36,40,51]. These components induce a story structure that is modeled as a partially ordered plan. The basic idea is to pre-compute every way the player can violate the plan and generate a contingency plan. The collection of all contingency plans and the narrative plan form the *narrative mediation tree*. To prevent unbounded mediation trees, certain player actions are surreptitiously replaced with "failures." This is similar to the "boundary violations" discussed by Magerko in the context of IDA.

The initial narrative plan represents the author's ideal story. In this sense, narrative mediation is similar to prototype based TTD-MDPs. It can be proven that this method of authoring interactive narrative is equally as powerful as creating branching story graphs.

Riedl & Stern implement this approach for a cultural training simulation [37,38,39]. This believable agent architecture, known as the *Automated Story Director* (ASD), has two goals: first, it must provide instruction to autonomous believable characters that help to shape the player's experience in the neighborhood around the narrative training goals; and second, it must monitor the story world to detect any inconsistencies that arise as a result of player actions and repair the narrative plan accordingly. To accomplish this, they modify the "failure" semantics discussed above to change the narrative goals of the system rather than simply fail.

This system shares a lot in common with IDA and Mimesis. If you consider the range from reactive to proactive enclosed by Mimesis on one end and IDA on the other, then ASD lives somewhere in the middle. ASD also shares some similarities with the beat-based drama manager of Mateas & Stern; however, in contrast to beat-based systems where non-determinism and loosely specified authorial goals provide

[4] In the work described here, the author constructs the model by hand. Mott, Lee & Lester have worked on predicting player goals by learning probabilistic models [28].

distinct player autonomy appropriate for narrative situations, this system uses a planning based approach to "recover" authorial goals when player actions change the training narrative flow. The ASD approach is well suited to training or learning environments where player autonomy is intended to support exploratory learning rather than improve the quality of the entertainment experience.

ASD, like U-Director, is **theoretically sound**. To our knowledge, this is the only system for which theoretical properties explicitly pertaining to authorial issues have been proven. Additionally, the handling of player autonomy is laudable, because contingencies for achieving authorial goals are *modus operandi*. On the other hand, the only source **replayability** comes from player choices.

In addition to ASD, the Mimesis system (see Section 5.2) also performs narrative mediation. Whereas ASD uses a completely pre-specified narrative mediation approach where all contingency plans are computed in advance, Mimesis accomplishes this through a complicated caching and speculative re-planning scheme. The Mimesis approach requires that all characters in the game (including the human player) obtain permission form the mediator before executing their actions. Thus, rather than fully determining all contingency plans, Mimesis can cache those most relevant to the current narrative plan and construct new ones as player actions move the narrative toward parts of the story space not as heavily represented by mediation plans in the cache.

## 6. Non-Planning and Non-Optimization Systems

In this section, we evaluate a number of approaches to drama management that do not fall under the classification of either an optimization-based system or a planning-based system. The technical approaches underlying these systems vary greatly, ranging from probabilistic graphical models to case-based reasoning.

### 6.1 U-Director

Mott & Lester developed *U-Director*, a narrative planning infrastructure that is designed to deal with the uncertainty in narrative environments induced by player autonomy [31]. Their goal is to develop a system that satisfies what they call narrative rationality. They define *narrative rationality* as reasoning in a principled manner about narrative objectives, story world state, and user state in the face of uncertainty to maximize narrative utility.

The "director agent" ensures plot progress and narrative consistency using *dynamic decision networks* (DDNs). DDNs are a generalization of Bayesian networks that include utility and choice nodes as well as time-varying attributes. The network is constructed using a level of abstraction similar to that of SBDM, DODM, and TTD-MDPs where DM actions are abstract directions that can have any number of concrete implementations in the game world.

They define a narrative decision cycle that is characterized by three levels of a dynamic decision network: the current game state (characterized by a decision node); the game state after the director's action has been taken (characterized by a chance node); and the game state after the player's reaction (characterized by a utility node). The utility nodes represent authorial intent in much the same way that the evaluation function does for SBDM and DODM. Each of these levels of the network contains nodes that represent details about the game and the players. With the exception of the work of Sharma *et al*. (see Section 6.4), this is the only drama management system surveyed that explicitly models the *player's goals*. The decision network contains nodes for the player's goals and beliefs (or knowledge gained about the salient facts of the story through interaction) as well as experiential state (or degree the player has been manipulated by the DM and how engaged they are in driving the plot). To actually make a decision, the director updates the narrative state according to the structure of the network in each of the three time slices associated with the current decision cycle. With the network updated, the director can perform action selection by analyzing each action's influence on the utility node in the third time slice.

In their tests, they have a network with 200 chance nodes, 400 causal links, and 7,000 conditional probabilities and a separate network of 50 nodes to express narrative utility preferences. It seems unlikely that the non-expert will find this **easy to author**; however, this approach is theoretically well-grounded in the body of work on dynamic decision networks and so is quite **sound**.

### 6.2 Beat-Based DM

Mateas & Stern define a narrative to be a sequence of events that induce "changes in values." These values are properties of individuals or relationships such as love, hope, or anger. They define a *beat* as the "smallest unit of value change" and a *scene* as a "large-scale story event" [23]. Computationally, a scene in an interactive narrative is defined by a number of annotations: a set of preconditions; the values that are changed during the scene; a large collection of beats to effect the desired change in values; and a temporal description of how the values should be changed during the scene. Thus, an interactive narrative is defined by a set of scene definitions.

With scenes as the basic building blocks, Mateas & Stern develop a *beat-based* drama manager and implement it in their interactive fiction Faç ade [24,25,26,27]. The drama manager is provided with a desired *global plot arc* that defines the shape of the change of the dramatic variables. The DM first determines the set of scene definitions that have satisfied preconditions and selects the one that matches the current position of the global plot arc as closely as possible. Then, the DM maintains a bag of beats associated with the current scene and reactively applies them until the desired value changes for the scene has been realized. Note that the change on dramatic values by a particular beat is a function of the beat's characteristics and the human player's participation. Thus, beats actually define an expectation over value change.

This authorial idiom is unique among all of the drama management systems surveyed in this paper. Due to the level of granularity required to author beats and their interactions, a beat-based drama manager seems ideally suited to the small-world variety of dramas like Faç ade; however, the freedom of **replayability** and **authorial control** may come at the price of **ease of authoring**, at least for large systems.

### 6.3 OPIATE

Fairclough implements a narrative story generation system called OPIATE. OPIATE uses a *story director* to drive narrative events in an open environment where the story is generated in real-time in response to the changing game environment and the player's actions [12]. The story director has a "world view" about the state of the game, using that to construct plans to achieve dramatic goals. It uses a *case-based planner* that is endowed with a plan library created using expert knowledge of skeletal plot structures and how they fit into the story world.

The cased-based planner uses its dramatic goals and plan library to synthesize plot-based and character-based stories. A k-nearest neighbor algorithm is used for case retrieval that additionally provides a "suitability" score for each of the retrieved cases—the most suitable case is the sub-plot that should be enacted given the current state of the story world and the current state of the characters (including their attitudes toward each other and the player). A "suitability threshold" is used to determine if the best case should be used or cases should be combined to create a new case to be enacted by the story director. The suitability score can be decomposed to provide an individual score for each "function" in the case. Thus, case combination is simply a matter of finding the highest scored set of functions and combining them to form a new case. Once a case is selected, a "casting" approach is used where the abstract instructions of the case are assigned to specific characters based on defined roles. For example, if the role of "hero" is embodied by the player, then the NPC that opposes the player the most will be cast as the "villain." Thus, as the relationships between the characters change throughout the dramatic experience, the cases that are retrieved change based on the suitability of the casting of the characters based on their relationships. This is similar to the work of Mateas & Stern on beat-based drama management where the scenes that are selected by the DM are chosen based on their fit to the dramatic values that represent the characters and their relationships.

There is notable **authorial effort** required to construct a case-base for the OPIATE system. On the other hand, its unique approach to dynamically casting non-player characters into different roles based on evolving relationships encourages **replayability** and provides a unique form of **coordination**

### 6.4 Player Preferences

.

Sharma *et al.* have taken a new approach to drama management by explicitly including a model of player preference in the DM's decision making [43]. Drawing a distinction between *player preference* models and *player action* models, they identify one potential criticism of many other methods (with the notable exceptions of Beat-Based DM and OPIATE): drama management techniques overwhelmingly use artificial models of player behavior that do not explicitly represent the players preferences or goals.

This approach is based on a simplification of the SAS+ algorithm that nonetheless extends it by combining the author's evaluation of a story and the player's preference for that story. They employ a *case-based reasoning* (CBR) system to determine player preferences by comparing their behavior to the behavior of earlier players. Preferences are elicited through a series of evaluation questions after an episode of game play. The weights on the player preference term and the author evaluation term in the heuristic function are adjusted depending on the "confidence" of the system that it has an accurate model of player preferences. Thus, if the system is able to confidently identify the current player as having a particular preference, it will guide her toward the types of the stories she enjoys; otherwise, it will attempt to preserve author intent.

Several issues arise. First, the author's evaluation function must be defined over partial stories. Nelson & Mateas have previously discussed the difficulties in authoring evaluation functions that are well defined in this manner [33,34]. Second, the particular choice of questions used for elicitation can be a cause for concern especially when the user is not completely sure of what she wants. Finally, it is unclear if the distinction between player preference models and player action models is necessary: explicitly modeling player preference may not provide increased representational power over implicitly modeling player preferences through the detailed modeling of their actions.

In any case, this system makes explicit the trade-off between **player autonomy** and **authorial control**. Further, the case-based approach is well-suited for online **adaption**. Of course, as with all learning techniques, CBR may require many examples to be effective, so extracting a player model may be difficult in practice. Insofar as this is difficult, the system reverts to SAS+. Insofar as this is possible, the system cedes authorial control.

## 7. Coordination Outside of Interactive Drama

Although the focus of this paper is on drama management in interactive entertainment, we feel that efforts in applying such techniques in other domains are instructive. In this section, we briefly mention *narrative-based learning* and *game balancing*.

### 7.1 Narrative-Based Learning

Recently, there has been growing interest in the use of games for instructional purposes. In educational and training environments, the teacher plays the role that the author plays in entertainment settings. Thus, the task of dynamically constructing engaging learning experiences in games is similar to the task of ensuring authorial intent in interactive narrative environments. Mott *et al*. have developed a multi-level planning architecture for narrative-based learning environments [29,30,32]. Ultimately, the goal of their system is twofold. First, the system must support the hypothesis-generation-testing cycles that are the foundation of exploratory learning. Second, the system must provide appropriate levels of motivation and engagement for the learner to succeed.

Their system uses two *hierarchical task network* (HTN) planners that operate at two levels of abstraction. The tutorial planner constructs plans that reflect the educational goals of the teacher. The narrative planner determines how best to carry out the tutorial plans at the concrete game level. Tutorial plans constrain the plan space of the narrative plans.

Mott *et al*. describe their HTN-based system as providing an intuitive and **easy authorial idiom**; however, their deterministic planning approach reduces **replayability**.

In addition to the work of Mott *et al*., Riedl *et al*. have also applied their work on ASD to training scenarios (see Section 5.3 for the discussion of that work).

### 7.2 Game Balancing

At a high level, drama management shares something in common with *dynamic game balancing*. That is, both game balancing agents and drama managers are tasked with making changes to the game world that will affect the player's experience. As discussed throughout this paper, the drama manager is generally designed to ensure authorial intent; however, a game balancing agent tries to modify the game world to ensure maximal enjoyment by the player. In that sense, the work of Sharma *et al.* on player preference modeling has elements in common with game balancing approaches as well as drama management approaches.

A frequently discussed example of game balancing is that of a first person action game. The more frequently the game is played, the more skilled the player will become at the combinations of button presses and timing required to master the game. As the player's skill level increases, it is likely the game will become less challenging and potentially cause the player to lose interest; however, if the game's difficulty is adjusted to keep the the player from mastering it, the player may also loose interest due to feeling like they are not improving. Traditionally, games have a static balancing component in the form of level selection (*e.g.* easy, medium, hard, or expert). Recent AI research applied to game balancing has given rise to the field of dynamic game balancing where the traditional "discrete" balancing through explicit player selection is replaced with intelligent game adaption and replayability across game episodes.

Our treatment of dynamic game balancing is brief due to space limitations; however, it is a rich area that supports a number of approaches, including reinforcement learning [1,2,3,4,5]; parameter manipulation [13]; dynamic scripting [44]; and genetic algorithms [11].

## 8. Discussion

We have surveyed a variety of systems for drama management in interactive drama. We have proposed a number of desiderata, including **speed, coordination, replayability, authorial control, player autonomy, ease of authoring, adaptability, soundness, invisibility, and measurabilty.**

The systems we have explored each have strengths; however, they all share common weaknesses. The approaches to drama management explored here have been focused on developing systems that provide some level of fidelity to the author's intent given a model of that intent; however, there is little evidence to suggest that any of the models proposed here are transparent to the typical author, who will presumably be an expert in narrative, but not in optimization, planning or any specific AI technique.

Recall that we have assumed that our hypothetical authors have created a pleasing narrative; however, it is unclear whether enforcing that author's narrative yields the most satisfying game play. Even if we can reasonably assume this problem away, it still remains to demonstrate that we have created systems that allow even highly motivated authors to express such narratives.

As such, we propose that future work focus on both the technical details of developing new frameworks for ensuring authorial intent and the user and ethnographic studies necessary to understand whether we have provided the proper authorial tools that allow designers to take advantage of our frameworks.

## 9. Acknowledgments

This research was performed while on appointment as a U. S. Department of Homeland Security (DHS) Fellow under the DHS Scholarship and Fellowship Program, a program administered by the Oak Ridge Institute for Science and Education (ORISE) for DHS through an interagency agreement with the U. S Department of Energy (DOE). ORISE is managed by Oak Ridge Associated Universities under DOE contract number DE-AC05-06OR23100. All opinions expressed in this paper are the author's and do not necessarily reflect the policies and views of DHS, DOE, or ORISE. We also acknowledge the support of DARPA under contract No. HR0011-06-1-0021.

## References

[1] G. Andrade, G. Ramalho, A. Gomes, and V. Corruble. Challenge-sensitive game balancing: an evaluation of user satisfaction. In Proceedings of the 4rd Brazilian Workshop on Computer Games and Digital Entertainment (WJogos05), pages 66–76, 2005.

[2] G. Andrade, G. Ramalho, A. Gomes, and V. Corruble. Dynamic game balancing: an evaluation of user satisfaction. In Proceedings of the 2nd Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE06). AAAI Press, 2006.

[3] G. Andrade, G. Ramalho, H. Santana, and V. Corruble. Challenge-sensitive action selection: an application to game balancing. In Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT05), pages 194–200. IEEE Computer Society, 2005.

[4] G. Andrade, G. Ramalho, H. Santana, and V. Corruble. Extending reinforcement learning to provide dynamic game balancing. In D. W. Aha, H. Muñoz-Avila, and M. van Lent, editors, In Proceedings of the 2005 IJCAI Workshop on Reasoning, Representation, and Learning in Computer Games, pages 7–12, 2005.

[5] G. Andrade, H. Santana, A. Furtado, A. Leitão, and G. Ramalho. Online adaptation of computer games agents: A reinforcement learning approach. Scientia, 15(2), 2004.

[6] J. Bates. Virtual reality, art, and entertainment. Presence: The Journal of Teleoperators and Virtual Environments, 2(1):133–138, 1992.

[7] S. Bhat, D. L. Roberts, M. J. Nelson, C. L. Isbell, and M. Mateas. A globally optimal algorithm for ttd-mdps. In Proceedings of the Sixth International Joint Conference

on Autonomous Agents and Multiagent Systems (AAMAS07), 2007.
[8] A. S. Cantino, D. L. Roberts, and C. L. Isbell. Autonomous nondeterministic tour guides: Improving quality of experience with ttd-mdps. Technical Report GIT-IIC-07-02, School of Interactive Computing, College of Computing, Georgia Institute of Technology, 2007.
[9] A. S. Cantino, D. L. Roberts, and C. L. Isbell. Autonomous nondeterministic tour guides: Improving quality of experience with ttd-mdps. In Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS07), 2007.
[10] D. Christian and R. M. Young. Comparing cognitive and computational models of narrative structure. Technical Report TR03-001, Liquid Narrative Group, Department of Computer Science, North Carolina State University, 2003.
[11] P. Demasi and A. Cruz. Online coevolution for action games. In Proceedings of The 3rd International Conference on Intelligent Games And Simulation, pages 113–120, 2002.
[12] C. Fairclough. Story Games and the OPIATE System. PhD thesis, University of Dublin - Trinity College, 2006.
[13] R. Hunicke and V. Chapman. Ai for dynamic difficulty adjustment in games. In Challenges in Game Artificial Intelligence AAAI Workshop, pages 91–96, 2004.
[14] L. P. Kaelbling, M. L. Littman, and A. P. Moore. Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 4:237–285, 1996.
[15] J. E. Laird, A. Newell, and P. S. Rosenbloom. Soar: An architecture for general intelligence. Artificial Intelligence, 33(3):1–64, 1987.
[16] A. Lamstein and M. Mateas. A search-based drama manager. In Proceedings of the AAAI-04 Workshop on Challenges in Game AI, 2004.
[17] B. Laurel. Toward the Design of a Computer-Based Interactive Fantasy System. PhD thesis, Drama department, Ohio State University, 1986.
[18] B. Magerko. Story representation and interactive drama. In 1st Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE05), 2005.
[19] B. Magerko and J. E. Laird. Building an interactive drama architecture. In First International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE03), pages 226–237, 2003.
[20] B. Magerko and J. E. Laird. Mediating the tension between plot and interaction. In AAAI Workshop Series: Challenges in Game Artificial Intelligence, pages 108–112, 2004.
[21] B. Magerko, J. E. Laird, M. Assanie, A. Kerfoot, and D. Stokes. Ai characters and directors for interactive computer games. In 16th Innovative Applications of Artificial Intelligence Conference (IAAI04), pages 877–883, 2004.
[22] M. Mateas. An Oz-centric review of interactive drama and believable agents. In M. Woodridge and M. Veloso, editors, AI Today: Recent Trends and Developments. Lecture Notes in AI 1600. Springer, Berlin, NY, 1999. First appeared in 1997 as Technical Report CMU-CS-97-156, Computer Science Department, Carnegie Mellon University.
[23] M. Mateas and A. Stern. Towards integrating plot and character for interactive drama. In Working notes of the Social Intelligent Agents: The Human in the Loop Symposium. AAAI Fall Symposium Series, 2000.
[24] M. Mateas and A. Stern. Architecture, authorial idioms and early observations of the interactive drama Faç ade. Technical Report CMU-CS-02-198, Carnegie Mellon Univeristy, 2003.
[25] M. Mateas and A. Stern. Faç ade: An experiment in building a fully-realized interactive drama. In Game Developer's Conference: Game Design Track, 2003.
[26] M. Mateas and A. Stern. Integrating plot, character, and natural language processing in the interactive drama Faç ade. In Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE-03), Darmstadt, Germany, 2003.
[27] M. Mateas and A. Stern. Structuring content in the facade interactive drama architecture. In Proceedings of Artificial Intelligence and Interactive Digital Entertainment (AIIDE05), 2005.
[28] B. Mott, S. Lee, and J. Lester. Probabilistic goal recognition in interactive narrative environments. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI06), 2006.
[29] B. Mott and J. Lester. Narrative-centered tutorial planning for inquiry-based learning environments. In Proceedings of the 8th International Conference on Intelligent Tutoring Systems (ITS06), 2006.
[30] B. W. Mott, C. Callaway, L. Zettlemoyer, S. Lee, and J. Lester. Towards narrative-centered learning environments. In Proceedings of the AAAI Fall Symposium on Narrative Intelligence, pages 78–82, 1999.
[31] B. W. Mott and J. C. Lester. U-director: a decision-theoretic narrative planning architecture for storytelling environments. In Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS06), pages 977–984, New York, NY, USA, 2006. ACM Press.
[32] B. W. Mott, S. W. McQuiggan, S. Lee, S. Y. Lee, and J. C. Lester. Narrative-centered environments for guided exploratory learning. In Proceedings of the Agent Based Systems for Human Learning Workshop at the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (ABSHL06), 2006.
[33] M. J. Nelson and M. Mateas. Search-based drama management in the interactive fiction anchorhead. In Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2005), 2005.
[34] M. J. Nelson, M. Mateas, D. L. Roberts, and C. L. Isbell. Declarative drama management in the interactive fiction anchorhead. IEEE Computer Graphics and Applications (Special Issue on Interactive Narrative), 26(3):30–39, 2006.
[35] M. J. Nelson, D. L. Roberts, C. L. Isbell, and M. Mateas. Reinforcement learning for declarative optimization-based drama management. In Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS06), 2006.
[36] M. O. Riedl, C. Saretto, and R. M. Young. Managing interaction between users and agents in a multi-agent

storytelling environment. In Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi Agent Systems, 2003.

[37] M. O. Riedl and A. Stern. Believable agents and intelligent scenario direction for social and cultural leadership training. In Proceedings of the 15th Conference on Behavior Representation in Modeling and Simulation (BRIMS06), 2006.

[38] M. O. Riedl and A. Stern. Believable agents and intelligent story adaptation for interactive storytelling. In Proceedings of the 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE06), 2006.

[39] M. O. Riedl and A. Stern. Failing believably: Toward drama management with autonomous actors in interactive narratives. In Proceedings of the 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE06), 2006.

[40] M. O. Riedl and R. M. Young. From linear story generation to branching story graphs. IEEE Computer Graphics and Applications, Special Issue on Interactive Narrative, 26(3), 2006.

[41] D. L. Roberts, A. S. Cantino, and C. L. Isbell. Player autonomy versus designer intent: A case study of interactive tour guides. In Proceedings of the Third Conference and Artificial Intelligence for Interactive Digital Entertainment (AIIDE-07), 2007.

[42] D. L. Roberts, M. J. Nelson, C. L. Isbell, M. Mates, and M. L. Littman. Targeting specific distributions of trajectories in mdps. In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06), 2006.

[43] M. Sharma, S. Ontaón, C. Strong, M. Mehta, and A. Ram. Towards player preference modeling for drama management in interactive stories. In Proceedings of the Twentieth International FLAIRS Conference (FLAIRS07), 2007.

[44] P. Spronck, I. Sprinkhuizen-Kuyper, and E. Postma. Difficulty scaling of game ai. In Proceedings of the 5th International Conference on Intelligent Games and Simulation, pages 33–37, 2004.

[45] M. van Lent, M. O. Riedl, P. Carpenter, R. McAlinden, and P. Brobst. Increasing replayability with deliberative and reactive planning. In Proceedings of the 1st Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE05), 2005.

[46] P. Weyhrauch. Guiding Interactive Drama. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1997. Technical Report CMU-CS-97-109.

[47] R. M. Young. Notes on the use of plan structures in the creation of interactive plot. In The Working Notes of the AAAI Fall Symposium on Narrative Intelligence, 1999.

[48] R. M. Young. An overview of the mimesis architecture: Integrating intelligent narrative control into an existing gaming environment. In Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, 2001.

[49] R. M. Young, J. D. Moore, and M. Pollack. Towards a principled representation of discourse plans. In Proceedings of the Sixteenth Conference of the Cognitive Science Society, 1994.

[50] R. M. Young and M. Riedl. Towards an architecture for intelligent control of narrative in interactive virtual worlds. In Proceedings of the International Conference on Intelligent User Interfaces, 2003.

[51] R. M. Young, M. O. Riedl, M. Branly, and A. Jhala. An architecture for integrating plan-based behavior generation with interactive game environments. Journal of Game Development, 1(1):54–70, 2004.