

Another Look at Search-Based Drama Management*

Mark J. Nelson

College of Computing
Georgia Institute of Technology
mnelson@cc.gatech.edu

Michael Mateas

Computer Science Department
University of California, Santa Cruz
michaelm@cs.ucsc.edu

Abstract

A drama manager (DM) monitors an interactive experience, such as a computer game, and intervenes to shape the global experience so it satisfies the author's expressive goals without decreasing a player's interactive agency. In declarative optimization-based drama management (DODM), the author declaratively specifies desired properties of the experience; the DM optimizes its interventions to maximize that metric. The initial DODM approach used online search to optimize an experience-quality function. Subsequent work questioned whether online search could perform well in general, and proposed alternative optimization frameworks such as reinforcement learning. Recent work on targeted trajectory distribution Markov decision processes (TTD-MDPs) replaced the experience-quality metric with a metric and associated algorithm based on targeting experience distributions. We argue that optimizing an experience-quality function does not destroy interactive agency, as has been claimed, and that in fact it can capture that goal directly. We further show that, though apparently quite different on the surface, the original search approach and TTD-MDPs actually use variants of the same underlying search algorithm, and that offline cached search, as is done by the TTD-MDP algorithm, allows the search-based systems to achieve similar results to TTD-MDPs.

Introduction

Interactive drama is a type of interactive experience in which a player interacts with a rich story world and experiences both a strong feeling of interactive agency and a dramatic, interesting, and coherent narrative. Giving a player a large degree of freedom in a story world and populating it with believable agents will not necessarily emergently create interactive drama, since interactions must not only be believable but also combine to form a globally coherent and interesting narrative.

A drama manager (DM) coordinates and adapts the agents and other contents of a story world as an experience unfolds in order to maintain global narrative coherence, without removing the player's interactive agency. In one approach, declarative optimization-based drama management (DODM), the author specifies the narratively important events, called *plot points*, that can occur in an experience. Examples of plot points include a player gaining

story information, changing their relationship with a non-player character, acquiring an important object, and so on. The plot points are annotated with ordering constraints that capture the physical possibilities of the story world. For example, events in a locked room are not possible until the player gets the key. Plot points are also annotated with bits of information that may be useful to the DM, such as where the plot point happens, what subplot it's part of, and so on (the exact set of annotations is flexible and depends on the story). Figure 1 gives an example set of plot points and ordering constraints.

The author also specifies a set of *DM actions* that the DM can take to intervene in the unfolding experience. Actions can *cause* specific plot points to happen, *hint* in a way that makes it more likely a plot point will happen, *deny* a plot point so it cannot happen, or *undeny* a previously denied plot point. For example, the DM might tell a non-player character to proactively approach the player and reveal some information, thereby causing the plot point associated with the player gaining that information. The set of plot points and DM actions, when combined with a player model, provide an abstract, high-level model of the unfolding experience.

Within this abstract view of interactive drama, the DM needs a way to choose actions. DODM has the author declaratively specify an optimality criterion; the DM then uses some method to choose optimal actions. DODM systems differ both on the conceptual issue of how to define optimality, and on the technical issue of how to carry out the optimization. The original DODM system used a game-tree-like search to maximize an evaluation function in which the author encodes a measure of experience quality (Bates 1992; Weyhrauch 1997). More recent work questioned both the feasibility of search as the optimization method (Nelson & Mateas 2005), and the idea of having the DM maximize an experience-quality function in the first place (Roberts *et al.* 2006). In particular, Targeted Trajectory Distribution Markov Decision Processes (TTD-MDPs) have proposed both a new goal of targeting an author-specified distribution of experiences, and associated algorithms to do so (Roberts *et al.* 2006; Bhat *et al.* 2007).

We revisit these criticisms. We argue that optimizing an experience-quality function rather than targeting an experience distribution does not destroy player agency as previously claimed, and that to the contrary, a well-written experience-evaluation function can directly target goals such as interactive agency, whereas simply adding nondetermin-

*This research was supported in part by a grant from Intel.
Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

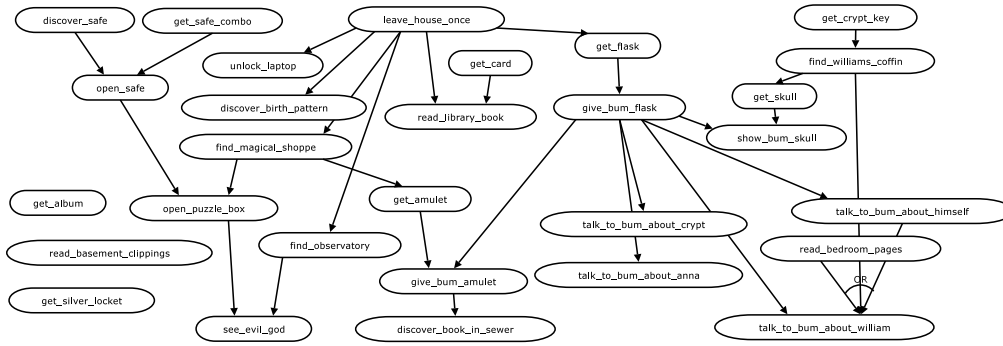


Figure 1: Plot points with ordering constraints for part of the interactive fiction *Anchorhead*, from Nelson & Mateas (2005).

ism via TTD-MDPs does not. As a technical matter we show that, although the algorithms appear different as originally described, the search-based optimization and the algorithm used by TTD-MDPs are variants of the same underlying search algorithm. Furthermore, when the original search algorithm is enhanced by caching, as the TTD-MDP one is, it performs at the same level.

Overview of DODM variants

DODM was proposed as search-based drama management (SBDM) by Bates (1992), and developed, implemented, and tested by Weyhrauch (1997). They conceived of DM by analogy to game-tree search: a player makes “user moves” through interaction with the game world (what we call *plot points*), and the DM responds with its own “system moves” (what we call *DM actions*). The DM chooses “moves” by running a game-tree-like search to maximize an evaluation function in which the author encodes her idea of a good interactive experience.

Since the DM process isn’t well modeled by an adversarial zero-sum game—a player is not actively trying to *minimize* the author’s evaluation function—SBDM uses an expectimax search rather than the minimax search used in adversarial games. In this expectimax search, the search tree alternates maximizing over the available DM actions with taking an expected value over the possible plot points that could follow, distributed according to a model of likely player behavior. In Weyhrauch’s system (and subsequent ones so far), a simple player model is used: the player is assumed to be equally likely to make each of the next possible plot points happen, except for those which have been hinted at, which are considered more likely by a multiplier that the author specifies in an annotation to the hint.

Weyhrauch found that a sampling search he developed, SAS+, was able to use the DM actions nearly optimally, as measured by the author-specified evaluation function and the player model. Loyall (2004) later reported some success using SBDM as part of *The Penguin Who Wouldn’t Swim*, a commercial prototype designed by Zoesis Studios.

Nelson & Mateas (2005) applied SBDM to a different story, modeled on a subset of the interactive fiction *Anchorhead*. They found that Weyhrauch’s excellent results with

sampling search didn’t transfer to their combination of plot points, evaluation function, and DM actions.

In response to this failure, Nelson *et al.* (2006) used reinforcement learning (RL) for the optimization step instead of search. Since RL runs offline to precompute a policy for using DM actions, it has the advantage of using offline computing time instead of having to make decisions during the much more limited time available during actual gameplay. However they found that in the same *Anchorhead* model, RL didn’t do well either, leading them to conclude that the DM actions specified weren’t sufficient to have much positive impact on the story, as measured by the given evaluation function—an authorship rather than optimization issue.

To test the optimization separately, they defined a “synthetic” set of DM actions, consisting of a causer, denier, and reenabler for every possible plot point. This was intended to provide a maximally powerful set of actions, regardless of whether it in practice could be readily implemented, in order to test the hypothetical success of various optimization methods on the pure optimization problem. They found that in this case, RL did quite well, in contrast to online search, which continued to perform relatively poorly.

Roberts *et al.* (2006) proposed a more fundamental change. They argued that when maximizing an evaluation function, the only source of gameplay variation will be the unpredictability of the player—and that given sufficiently powerful DM actions, the system could force its idea of optimal story on the player, destroying the interactive agency of the experience. They therefore proposed to start with a desired distribution of experiences (trajectories through the story space), and aim to use the DM actions in a way that would make the actual distribution come as close to the target distribution as possible. Algorithmically, the TTD-MDP system builds a large tree sampled from the space of all possible trajectories; each node in the tree then solves an optimization problem to find a distribution over its available actions that will, according to the player model, cause the resulting distribution over successor plot points to come as close as possible to the distribution specified by the author.

What to optimize

The fundamental conceptual issue in drama management is deciding what constitutes a good interactive drama. Given

criteria for good interactive drama, we can design the DM to try to bring about such an experience.

Formal and material constraints

Mateas (2000) proposes a theory of interactive drama integrating Aristotelian dramatic theory with Murray's (1998) desiderata for interactive stories, in particular the goal of interactive agency. He proposes that good interactive drama achieves a good balance of *material constraints*—the “constraints from below” that literally constrain what a player can do—and *formal (plot) constraints*—the “constraints from above” that constrain in the player's mind what, given the experience so far, is interesting, sensible, or worth doing.

The desired balance can be illustrated by contrasts with the extremes. In open-world sandbox and puzzle-based adventure games, the player can take many actions, but there is little plot that would give a reason to do anything in particular or serve as an interpretive framework tying events together. In linearly scripted games, meanwhile, the plot unfolds in a coherent fashion and everything the player does relates to a coherent whole, but many actions that make sense given the story are not supported by the game.

The actions in DODM serve to tinker with the formal and material constraints. Hints add new formal constraints by giving the player some additional narrative framework, without adding any new material constraints (the player may ignore the hint). Causers also add new formal constraints by directly causing some narratively important event to happen, but do so by also adding material constraints, since they temporarily remove the player's freedom to decide what should or shouldn't happen next. Deniers, meanwhile, add new material constraints, which can later be removed by undeniers.

Maximizing experience quality

The original DODM approach has an evaluation function that, given a completed experience (a sequence of plot points and DM actions), rates it based on various features that the author thinks an experience should have. The evaluation function therefore should be written so that it specifies to the DM what constitutes an experience in which the formal and material constraints are balanced; with such a function, the DM can tweak the constraints using its DM actions.

Although the terminology has sometimes been used loosely, the evaluation function in DODM rates the quality of *interactive experiences*, not the quality of plot-point sequences considered as stories alone. That is, DODM does not create interactive drama by taking a set of desiderata for *non-interactive* drama and trying to bring it about in the face of interactivity. Rather, it tries to maintain a set of desiderata for the interactive dramatic experience itself. Some DM systems do describe the drama-management problem as mediation between authorial narrative goals and player freedom (Young *et al.* 2004; Magerko 2005), and that view has sometimes been proposed as the general goal of drama management (Riedl, Saretto, & Young 2003; Roberts & Isbell 2007). It is not however the way DODM systems have typically viewed the problem. Rather than starting with an author-desired narrative and working around the user to bring it about, the system instead starts with an

idea of what constitutes a narratively interesting experience, and dynamically adjusts the material and formal constraints in the story world in order to ensure that such an experience comes about—working with the player to jointly create the narrative (Weyhrauch 1997; Loyall 2004).

Weyhrauch's evaluation function specifies a number of weighted features that capture his notion of a good experience in his *Tea for Three* story world.

One group of features serves mainly to encourage narrative coherence—more formal constraints and, where necessary, material constraints, to keep the player on track. These features include *thought flow*, which prefers stories where subsequent actions relate to each other; *activity flow*, which prefers stories that have spatial locality of action; and *momentum*, which prefers certain pairs of plot points that build on each other particularly well. Separately, the *motivation* feature prefers stories in which plot points are motivated by previous ones, such as a plot point in a detective story motivated by earlier clues.

These are explicitly preferences for the interactive experience, and would not necessarily be the same if evaluating a linear story. It may not be bad for narratives to have the action move around frequently between different locations, but Weyhrauch argues that if each plot point happened in a different location from the last, that would likely indicate in an interactive experience that the player was getting stuck in boring wandering around the world between plot points.

Given only these features, there is a danger that the system could identify certain plot-point progressions as ideal and force the player into them, adding too many material constraints and reducing interactive agency. To avoid this outcome, two versions of an additional evaluation feature—one proposed by Weyhrauch and one by Nelson & Mateas—aim explicitly at encoding interactive agency, though from different perspectives.

Weyhrauch's *options* feature identifies twelve meaningful goals a player might have at various points in *Tea for Three*. The goal “talk to George about the new will” is considered to be active between the time the player finds a note mentioning a new will and the time that the player either talks to George about it or is prevented from doing so by other events. The number of goals active at any given time is a rough measure of the degree of interactive agency available. The *options* feature encodes a preference for many such meaningful options to be available towards the beginning of the game, decreasing to fewer towards the end.

Nelson & Mateas's *choices* feature looks at the issue bottom-up instead, measuring how many plot points could have followed each point in the story, considering the ordering constraints in the world and the effects of causers and deniers. This is a rough measure of how much freedom the player had to influence the direction of the story locally. If at some point only one plot point could possibly have come next (because the DM caused it directly, for example), then the same bit of story would have played out regardless of what the player did. If, on the other hand, many plot points could have come next, the player could locally influence the story to a much greater extent. The *choices* feature has the advantage that it can be computed automatically without

additional authored knowledge, but the *options* feature has the advantage that it captures a higher-level notion of meaningful interactive agency. Both features capture a notion of preferring stories that preserve player choice, demonstrating that this can be represented directly in the optimality criterion.

Finally, *manipulativity* penalizes uses of DM actions that are likely to be particularly noticeable, such as moving objects that the player can see. This is a meta-feature encoding a preference for the DM's operation to be unnoticed. Although we use agents in service of a narrative rather than merely simulating them as believable agents in their own right, we do still want them to remain believable.

Targeting an experience distribution

Roberts *et al.* (2006) criticize maximization of a story-quality function, arguing that if the DM is too effective, it will bring about the same highly-rated story each time, destroying interactive agency and replayability. They propose that the goal should be to target a distribution of experiences, specified either by some mapping from an evaluation function (*e.g.* bad experiences should never happen, and good ones should happen in proportion to their quality), or by having the author specify a few prototype experiences and then targeting a distribution over experiences similar but not identical to the prototypes (Roberts *et al.* 2007).

While this is a valid criticism of maximizing *story* quality, the goal of DODM is to maximize *experience* quality. If the experience quality criterion appropriately includes features related to player agency, such as *options* and *choices*, then the DM will not force the same story every time.

More problematically, by targeting a specific distribution of experiences, TTD-MDPs do not necessarily coerce the player any less than a hypothetical system that targets a specific maximum-quality story would. A story-maximizing system that does not take into account features for player agency could indeed directly cause its same top-rated story every time. The TTD-MDP system, given powerful causers and deniers, can achieve a distribution over experiences by directly coercing each particular play-through. In either case, the player of any particular story would have no interactive agency, since in both cases the system would use its DM actions to produce a specific story. The TTD-MDP system would change *which* story it was forcing the user into each time, but randomly selecting a different story to force the user into each time is not an improvement in interactive agency.

If we look at the DM actions performed by the TTD-MDP based system and the maximization-based system on the version of *Anchorhead* with a “synthetic” set of DM actions that Roberts *et al.* use as a point of comparison, we do indeed find a similar level of coerciveness. The “synthetic” set of actions consists of a causer, denier, and reenabler for every possible plot point in the story, thus giving the DM a maximally powerful set of actions. The TTD-MDP system claimed better replayability in this case, since it produced a wider variety of stories. However, both the TTD-MDP system and the search-based evaluation-function-maximization system acted almost maximally coercively: they each per-

formed an average of around 15 DM actions per experience, in an experience 16 plot-points long. The TTD-MDP system varied which specific coercion it performed from run to run, but that does not constitute interactive agency, which requires that the *player*, rather than system nondeterminism, be able to meaningfully influence the outcome.

That both systems are quite coercive does point to a failure in the experience-quality evaluation function that both used. We can correct this by putting a greater weight on the *choices* feature, emphasizing that giving the player choices in what to do really is an important part of an interactive experience. When we increase *choices* from being 15% of the total evaluation weight to 50%, both systems drop to using an average of around 5 DM actions per experience.

Thus maximizing an experience-quality function need not destroy an experience's interactive agency, if it appropriately rates highly only those experiences that do actually have good interactive agency. If an evaluation function *does* capture interactive agency well, then an evaluation “spike” in an evaluation-function graph like that in Figure 3 would not be a problem. The fact that a DM produced all high-quality experiences would not necessarily imply, as has previously been suggested (Roberts *et al.* 2006), that it was linearizing the experience and always making the same literal experience come about, since many different experiences may have similar high ratings.

How to write evaluation functions so that they really do capture interactive experience quality remains an issue that would benefit from more experimentation in specific real interactive dramas. It is worth noting that all the recent systems have focused on the “synthetic” model of *Anchorhead* that has only causers, deniers, and reenablers, and lacks the hint DM actions that a DM could use to add formal constraints by providing more narrative to the player without unduly removing interactive agency; by contrast, a real application would likely use hints frequently.

Whether the TTD-MDP formulation still improves matters in a different way depends on how the target distribution is defined, and on what we consider to be the goals of interactive drama. When the target distribution is generated by a mapping from an experience-quality function, the results will be fairly similar to the results from an evaluation-function-maximizing approach, since both systems will be trying to avoid low-rated experiences and increase the probability of highly-rated ones according to the same function. The TTD-MDP approach will add some more nondeterminism in doing so; how much depends on how the mapping is constructed. Alternate ways of specifying a target distribution of experiences for TTD-MDPs, however, such as specifying several prototype experiences and inducing a distribution over experiences similar to those prototypes (Roberts *et al.* 2007), suffer from a greater loss of interactive agency. If the player is being forced into one of several prototype experiences or minor variants, the fact that the specific experience they're forced into is chosen nondeterministically does not preserve interactive agency.

In either case, the nondeterminism of TTD-MDPs serves a different goal than that of interactive agency. Interactive agency requires that if a player does things *differently*, then

```

Build a large tree of possible experience trajectories
for all nodes  $n$  in a post-order (leaf-first) traversal do
  if  $n$  is terminal then
     $n.value \leftarrow \text{terminalValue}(n)$ 
  else
     $n.policy \leftarrow \text{optimalPolicy}(n.actions, n.children)$ 
     $n.value \leftarrow \text{backup}(n.children, n.policy)$ 
  end if
end for

```

Figure 2: Generic cached search. The TTD-MDP algorithm and expectimax search share this structure, but differ in how they define the three functions (see text).

they should be able to experience meaningfully different narratives. It does not require a player to experience different narratives even if they do the exact same thing, which is the goal at which TTD-MDPs aim. Indeed, some ideas for interactive narrative, such as the “kaleidoscopic narratives” discussed by Murray (1998), depend on a player being able to tease out how acting differently results in a different narrative, which nondeterminism is likely to obscure.

Non-dramatic interactive experiences

We focus on authoring interactive drama. Similar experience-management techniques can be used for experiences other than interactive drama, which may have different considerations. For example, we argue that in interactive drama, the drama-management problem is best seen as helping to ensure that there is enough narrative for the player to have a coherent and interesting experience. As Loyall (2004) describes it, the DM picks up the slack in creating narrative when the player would be lost, but lets the player drive the narrative otherwise.

Other experiences may have genuinely external constraints that could conflict with the user’s freedom and goals. For example, a TTD-MDP system was proposed for guiding museum tours (Cantino, Roberts, & Isbell 2007). In that domain, the goal of reducing congestion really is an external goal imposed on the visitors, and is reasonably expressed by targeting a specific distribution of experiences so as to keep visitors nicely spread out. Training scenarios may also have an externally imposed requirement that a particular distribution of desired situations be encountered over a series of training runs.

Optimization by cached search

The original search-based formulation used an online expectimax search that, to remain computationally tractable, switched to sampling after a depth limit. The TTD-MDP algorithm operates offline, sampling many possible trajectories through the story world and building them into a tree, and then solving an optimization problem at each node. When a trajectory is seen that wasn’t among those sampled in the tree, it falls back to online search.

While described differently, these algorithms are quite similar when expectimax search is extended to also use a tree of cached results. Both build a cached tree, perform

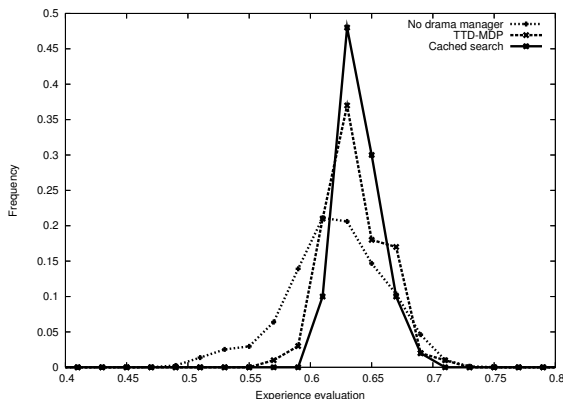


Figure 3: Frequency of experiences of various qualities for a simulated user with a DM guided by iterative-deepening search assuming a minute between plot points, TTD-MDP with a fallback to online sampling search, and a no-DM baseline.

an optimization at each node starting from the leaves and working upwards, and back results up the tree, as shown in the generic pseudocode in Figure 2. The main differences are that they choose actions at each node using a different objective function, and assign and back up values to nodes based on different evaluation criteria.

In expectimax search, terminal node values are given by the evaluation function. The policy at each node takes the DM action that maximizes expected evaluation value, given the player model. The node’s own value is this maximum expected value. In the TTD-MDP algorithm, the terminal node values are target probabilities. The policy at each node specifies a distribution over DM actions that results in the minimum expected divergence from the target experience distribution specified by the node’s children. The node’s own value is the sum of its children’s target probabilities.

Both algorithms can adaptively fill in their cache during gameplay, using background processor cycles to better fill in the tree rooted at the current point in the story (Bhat *et al.* 2007). In fact, once we note the connection with search, we can consider well-known space versus time tradeoffs to avoid having the cache at all. Maintaining a tree and filling in nodes at the frontiers is essentially breadth-first search, which has nice execution-time properties but exponentially large memory requirements. A common alternative is iterative deepening search, in which we perform depth-first, fixed-depth searches of increasing depth, stopping and returning the results of the deepest completed search when we need the next decision. This trades off a constant-factor increase in execution time for exponentially decreased memory requirements, allowing us to use a search-based DM on platforms with relatively little memory.

To demonstrate that iterative deepening search running in the background works effectively, Figure 3 shows histograms of the frequency with which experiences of varying evaluations appear over a number of runs with a simulated player (the same acting-randomly-except-for-hints player

used by previous work). The DM in this setup uses the “synthetic” set of DM actions consisting of a causer, denier, and reenabler for every possible plot point; this was the maximally powerful setup in which Nelson *et al.* (2006) found that search still could not work well. Of the three curves, one shows the results without a DM; one with a DM controlled by TTD-MDP, with a tree of 1.5 million trajectories and sampling search used as a fallback if a trajectory not in the tree is encountered; and the final one with a DM controlled by cached search. As can be seen from the shapes and positions of the histograms, cached search and TTD-MDP have almost identical performance, successfully avoiding the poorly rated experiences while boosting the frequency of highly rated experiences.

Reinforcement learning can also optimize well (Nelson *et al.* 2006), and constructs a compressed action policy offline using a function approximator. However, using function approximators with reinforcement learning in large and complex state spaces can be quite unstable, and requires significant tuning of free parameters, so we prefer cached search’s predictability and simplicity.

Conclusions and Future Work

We review variants of declarative optimization-based drama management (DODM), and defend a version of the formulation proposed by Bates (1992) and Weyhrauch (1997). We separate what to optimize from how to do so, and defend maximizing an experience-quality function, pointing out that experience-quality functions are not equivalent to *story*-quality functions, and do not rate experiences as if they were non-interactive narratives; rather, they explicitly take into account elements of a good interactive experience, such as interactive agency. We argue that TTD-MDPs, by contrast, primarily serve to add nondeterminism to their actions, which does not in itself produce interactive agency. On the technical issue of how to carry out optimization, we show that the tree-based algorithm used by TTD-MDPs and the search-based algorithm used by the original DODM are versions of a generic search-based algorithm, to which caching or offline computation may be added.

While we hope this clarifies the differences between existing DODM variants, we have only touched on the relationships between DODM and other DM systems. We’ve argued that unlike some systems that view drama management as a process of maintaining a narrative in the face of player interaction, DODM aims to ensure that a player has an interesting and coherent narrative experience without enforcing any particular one. Future work should compare the usefulness of various DM systems for authoring real interactive dramas.

There are elements that all DODM variants have held fixed that might be productively varied in future research. All use a simple player model that acts mostly randomly in the plot-point space. This ignores the fact that, for example, plot points near the player in the world are much more likely to happen than those not near the player. A simple improvement would be to model the player as acting randomly in a model of the physical layout of the story world, and use this to induce a distribution over plot points. That would

allow the story world’s layout to influence drama management, while still allowing the DM itself to operate purely on the abstracted story world.

The sequence of plot points and DM actions may also fail to capture important information. For example, the player may wander around the world for a long time without experiencing anything. The DM doesn’t notice any change in its abstracted view, so does nothing. One solution is to make lulls in action trigger a reusable “lull in action” plot point.

Finally, the ultimate point of any DM system is to be useful in building real interactive dramas. Therefore our future work will consist primarily of experimenting with many of our proposed improvements and evaluation function elements in the context of implemented, playable experiences.

References

- Bates, J. 1992. Virtual reality, art, and entertainment. *Presence* 1(1):133–138.
- Bhat, S.; Roberts, D.; Nelson, M. J.; Isbell, C. L.; and Mateas, M. 2007. A globally optimal algorithm for TTD-MDPs. In *Proc. AAMAS 2007*.
- Cantino, A.; Roberts, D. L.; and Isbell, C. L. 2007. Autonomous nondeterministic tour guides: Improving quality of experience with TTD-MDPs. In *Proc. AAMAS 2007*.
- Loyall, B. 2004. Response to Janet Murray’s ‘From game-story to cyber-drama’. In *First Person: New Media as Story, Performance, and Game*. MIT Press. 2–9.
- Magerko, B. 2005. Story representation and interactive drama. In *Proc. AIIDE 2005*.
- Mateas, M. 2000. A neo-Aristotelian theory of interactive drama. In *Proc. of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, 56–61.
- Murray, J. 1998. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. MIT Press.
- Nelson, M. J., and Mateas, M. 2005. Search-based drama management in the interactive fiction Anchorhead. In *Proc. AIIDE 2005*.
- Nelson, M. J.; Roberts, D. L.; Isbell, C. L.; and Mateas, M. 2006. Reinforcement learning for declarative optimization-based drama management. In *Proc. AAMAS 2006*.
- Riedl, M. O.; Saretto, C. J.; and Young, R. M. 2003. Managing interaction between users and agents in a multi-agent storytelling environment. In *Proc. AAMAS 2003*.
- Roberts, D. L., and Isbell, C. L. 2007. Desiderata for managers of interactive experiences: A survey of recent advances in drama management. In *Proc. Workshop on Agent-Based Systems for Human Learning and Entertainment at AAMAS*.
- Roberts, D. L.; Nelson, M. J.; Isbell, C. L.; Mateas, M.; and Littman, M. L. 2006. Targeting specific distributions of trajectories in MDPs. In *Proc. AAAI 2006*.
- Roberts, D. L.; Bhat, S.; St. Clair, K.; and Isbell, C. L. 2007. Authorial idioms for target distributions in TTD-MDPs. In *Proc. AAAI 2007*.
- Weyhrauch, P. 1997. *Guiding Interactive Drama*. Ph.D. Dissertation, Carnegie Mellon University.
- Young, R. M.; Riedl, M. O.; Branly, M.; Jhala, A.; Martin, R. J.; and Saretto, C. J. 2004. An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development* 1(1).