Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment

An Offline Planning Approach to Game Plotline Adaptation

Boyang Li and Mark O. Riedl

College of Computing, Georgia Institute of Technology
{boyangli, riedl}@gatech.edu

Abstract

Role-playing games, and other types of contemporary video games, usually contain a main storyline consisting of several causally related quests. As players have different motivations, tastes and preferences, it can be beneficial to customize game plotlines. In this paper, we present an offline algorithm for adapting human-authored game plotlines for computer role-playing games to suit the unique needs of individual players, thereby customizing gaming experiences and enhancing re-playability. Our approach uses an plan refinement technique based on partial-order planning to (a) optimize the global structure of the plotline according to input from a player model, (b) maintain plotline coherence, and (c) facilitate authorial intent by preserving as much of the original plotline as possible. A theoretical analysis of the authorial leverage and a user study suggest the benefits of this approach.

Introduction

In the entertainment industry, such as film or game production, a significant amount of time and financial cost is incurred during the process of content creation. In comparison, the consumption of content is usually much faster. For example, it takes much less time for game players to complete games, expansion packs, and new quests than they can be produced. The content creation bottleneck results from the high financial constraints of producing content and underscores the need for Artificial Intelligence (AI) techniques for *on-demand* and *uniquely customized* entertainment experience.

On-demand entertainment refers to the possibility that one can, at any time, request an entertainment experience that is significantly different from those previously consumed. In addition, entertainment artifacts should be customized or configured to suit every player's unique motivation, tastes, history and ability. Due to their cost, there cannot be enough expert content producers to cater to every single consumer. In this paper, we explore the use of automated content generation as a means of scaling up the delivery of on-demand, customized entertainment content.

This paper addresses on-demand and customized entertainment in the context of generating plotlines for role-playing games. Rollings and Adams (2003) argue that the core of gameplay in any game is "one or more causally linked series of challenges in a simulated environment." To overcome these challenges, players have to perform required gaming activities such as combat or puzzle-solving in a virtual world. Role-playing games in particular deliver challenges through *quests*. The set of quests that are necessary and sufficient to complete the game make up the main plotline of the game. Often side-quests are available to augment the gameplay experience, and to allow players a limited degree of customization through choice.

We argue that customization of main plotline involves presenting *the right story to the right person at the right time*. The significance of this claim is twofold. First, game players usually possess diverse motivation, tastes, and needs (Yee 2006), which a one-size-fits-all script might not meet. Moreover, to achieve optimal game experience, challenges must be adapted to the player's skill level. Second, preferences of players can change over time. Having played one story, the player may demand a new one. Therefore, the ability to generate customized plotlines may enhance replayability and improve player experience. By addressing the two implications, we are working toward the potential of games that continuously grow and change with the player over a long period of time by generating novel, customized plotlines.

In the realm of automated content generation, plotlines may be generated from scratch. Such ability has long been the goal of story generation. However, how to build computational systems with aesthetic capabilities rivaling that of human designers is still an open research question. Until we have systems of sufficient aesthetic capabilities to assume full responsibility for creating entire plotlines, we can consider starting with existing plotlines and automatically altering them to novel contexts according to knowledge about the player. Plotline adaptation has the advantage of exploiting human intuition to the extent possible without necessarily having a complete model of aesthetic reasoning. Plotline adaptation promises good authorial leverage (Chen et al. 2009) to produce a greater number of meaningful, customized, and novel interactive experiences for individual players with less effort.

The contribution of this paper is a system that solves the *plotline adaptation problem*: Given a complete, human-authored plotline consisting of sequence of quests, a library of quest structures, and a set of requirements about what an optimal gameplay experience should be for a particular user at a particular time, produce a sound, coherent variation that meets the requirements while retaining as much of the original plotline as possible. A sound plotline is one that, in the absence of uncertainty, is guaranteed to play out as intended. A coherent plotline is one that does not have any unnecessary elements. Finally, preservation of the original plotline prevents unnecessary modifications and preserves the human authors' intent to the extent possible while meeting other objectives. Offline processing affords the possibility to making globally optimal decisions about plotline structures, complimenting online adaptation techniques known as *interactive storytelling*.

The remainder of the paper is organized as follows: We first review relevant work in game adaptation and narrative generation. Next, we explain the representation of quests and the offline adaptation algorithm. Our technique is justified with a discussion on theoretical authoring gains and empirical evaluation of our algorithm.

Related Work

Automated adaptation of computer games has been explored in the context of player character attributes, difficulty adjustment, and game environment changes. Increasingly, player models are being used to adapt game content. Interactive storytelling systems demonstrate how players' behaviors can change the story content in virtual worlds on the fly. See Roberts and Isbell (2008) for a general discussion of interactive narrative approaches. Of particular relevance to this work are interactive narrative approaches that leverage player models. Thue et al. (2007) describe a technique whereby a player model based on role player types is used to select branches through an interactive story. Seif El-Nasr (2007) attempts to infer feature-vectors representing player style, affecting changes in which dramatic content is presented to the player. Sharma et al. (2010) use case-based reasoning to learn player preferences over plot points for the purposes of selecting the next best story plot point. These approaches assume the existence of branching story graphs or pre-authored alternatives.

Note that our system is an offline process that effectively "re-writes" a plotline based on a player model before it is executed. As such, our system can afford to backtrack and make globally optimal decision, such as those about narrative coherence, whereas online adaptation systems can only make local decisions that cannot be undone. Our system is *not* an interactive narrative system; once execution of the plotline begins, our system does not make further changes. Indeed, interactive storytelling and plotline adaptation are complimentary: the adaptation system can be seen as a process that, based on knowledge about the player, configures the drama manager, which then oversees the user's interactive experience online. Our system can be coupled with, for instance, the Automated Story Director (Riedl et al. 2008), a planning-based interactive narrative system.

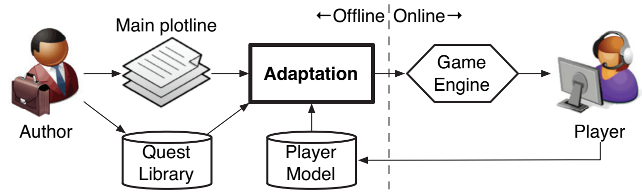

Figure 1. The plotline adaptation pipeline

As an offline procedure, plotline adaptation has a strong connection with story generation. Story generation is the process of automatically creating novel narrative sequences from a set of specifications. The most relevant story generation work is that that uses search as the underlying mechanism for selecting and instantiating narrative events (cf., Lebowitz 1987; Porteous and Cavazza 2009; Riedl and Young, in press). The distinction between our plotline adaptor and story generation is that plotline adaptation starts with a complete narrative structure and can both add and remove narrative content, whereas story generation typically starts from scratch. As with case-based planning, the adaptation of plotlines is, in the worst-case, just as hard as planning from scratch (Muñoz-Avila and Cox 2008). However, in the average case, starting from an existing plotline will require much fewer decisions to be made.

Game Plot Adaptation

Figure 1 shows the adaptation pipeline for game plotlines. The game adaptation process takes a main plotline, a library of quest structures, and a set of player requirements as input. There may be more than one human-authored plotline, of which one is provided as input into our system. A player model generates a set of requirements, based on the player's preferences, history, and a model of novelty. Importantly, the cycle between player and adaptation implies that games can be changed after each time it is played, thereby increasing its replayability. Possible actions in the virtual world are known to the system.

Plot Representation

Following others (Young 1999; Riedl et al. 2008; Porteous and Cavazza 2009), we employ plan-like representations of narrative because they capture causality and temporality of action and provide a formal framework built on first principles, such as soundness and coherence, for selecting and ordering events. However, unlike a plan meant for execution, we use plans as descriptions of events expected to unfold in a virtual world. Each action represents a formal declaration of an event that can be performed by the player or non-player characters, or occur as a consequence of physics laws in the virtual world.

Our specific representation builds on *partial-order plans*. A partial-order plan consists of events and temporal and causal relations. Events encode preconditions, which must be true for the event to occur, and effects, which become true once the event completes. Causal links, denoted as $e_1 \rightarrow^c e_2$, indicate that the effects of event e_1 establish a condition c in the world necessary for event e_2. Causal links act as protected intervals during which the truth of condition c in the world must be maintained. Temporal links indicate ordering constraints between events. Additionally, to capture semantic meaning of narrative subsequences, we allow for event abstraction hierarchies. Abstract actions are decomposed into sequences of equivalent, but less abstract events. Decomposition rules act as grammars specifying legal configurations of narrative fragments. Decomposition rules must be authored *a priori* and are one way to leverage human authorial intuition; partial-order planning may discover causal and temporal relations based on the rules.

In our system, *quests* are represented as top-level abstract events. A quest has a single effect, `quest-complete(quest-X)`, and may or may not have any preconditions. While not strictly necessary, we find the following authorial idiom to work well: decomposition rules break quests into an abstract *task* event and an abstract *reward* event, which are further decomposed into primitive events. Figure 2 shows a complete plotline consisting of two quests. Primitive actions are shown as solid rectangles and abstract actions are shown as rounded rectangles. The hierarchical relationship between events is reflected in the containment relationships of rectangles. For example, one legal way in which a `witch-hunt quest` can occur is to kill the witch with water and earn the trust king. Arrows represent causal links. Note that not all causal links are shown for clarity's sake. Temporal links are omitted.

Formally, on each hierarchical level, a plot can be viewed as a directed acyclic graph (DAG). A *sound narrative* is one in which all preconditions of an event are guaranteed to be true when it is scheduled to execute, and all abstract events are decomposed to the fullest extent. In other words, a sound narrative is one that does not violate the physics of the story world as defined by the preconditions of events and impossible world states. A *coherent narrative* is one in which, in the DAG formed by events and causal links, there is a path from each event to a significant outcome. Any event that is not part of some path to the outcome situation is referred to as a *dead end*. The concept of coherence and dead ends is a computational interpretation of a cognitive model of narrative comprehension by Trabasso and van den Broek (1985).

Player Model

We model the player's preference as a function of previously selected quests. Each quest, in turn, is represented as a feature vector in a semantic space. We utilize a technique similar to that in Sharma et al. (2010) to determine preferences over quests via ratings after gameplay concludes; similarity metrics allow us to extend preferences to quests not previously experienced by the user. In addition, a novelty model based on (Saunders and Gero 2004) favors quests that are appropriately novel to the player based on his or her history so that he or she would be neither bored nor unpleasantly surprised. Computing a weighted sum of utility by preference and utility by novelty, the result is the selection of the k quests with the greatest utility that should be included in the game plotline. Due to space constraints, a detailed description is beyond the scope of this paper.

Adaptation Algorithm

The plot adaptation module receives as input the following components:

- A complete plotline – a partially ordered, hierarchical plan – composed of events within and outside of quests.
- A set of plot requirements: `quest-complete(quest-X)` propositions specifying what quests should be included, and corresponding world-level outcome propositions.

The adaptation process involves two stages. In the first stage, a problem instantiation is created by rewriting the initial world state and desired outcome situation to match the plot requirements. When rewriting the outcome situation, any quests that no longer causally link to the

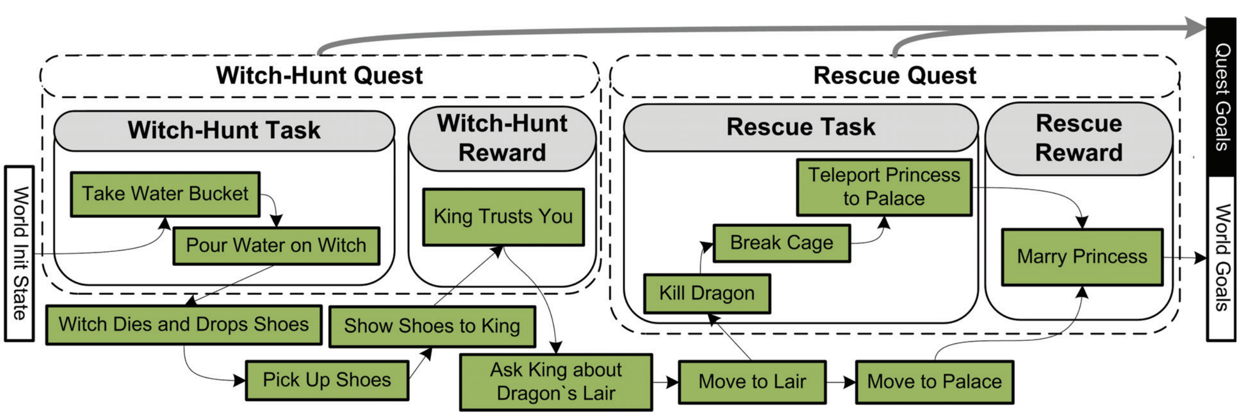

Figure 2. Original game plotline before adaptation

The algorithm takes a plotline plan, a set of rules to rewrite the goal and initial state, and a domain library Λ consisting of events specifications and quest decomposition rules.

```
function ADAPT (plan, reqs, Λ) returns solution or failure
  plan ← REWRITE-GOAL-AND-INITS(plan, reqs)
  fringe ← {plan}
  loop do
    if fringe = ∅ then return failure
    plan ← POP(fringe)
    if plan has no flaws then return plan
    flaw ← GET-ONE-FLAW(plan)
    newplans ← REPAIR(flaw, plan, Λ)
    fringe ← INSERT-AND-SORT(newplans, fringe)
```

Figure 3. The plotline adaptation algorithm

outcome situation become dead ends and the plotline is no longer coherent. When rewriting the initial state, the preconditions of some events may no longer be supported by the initial state and the plotline may no longer be sound.

The second stage is plan refinement search process that progressively makes adjustments to the plotline until (a) all plot requirements are met, (b) the plotline is sound, and (c) the plotline is coherent.

Plan refinement techniques search a space where each node in the space is an instance of a plan (partial or complete) until a plan is found that has no *flaws*, or reasons why a plan cannot be considered a solution. Partial-order planning (cf., Penberthy and Weld 1992) is a form of plan refinement search that starts with the empty plan. For each plan visited, a flaw is detected and all repair strategies are invoked, each strategy resulting in zero or more new plans in which that flaw has been repaired. These new plans are successors to the current plan and are added to the fringe of the search space. A heuristic is used to determine which plan on the fringe visit next. Note that repairing a flaw may introduce new flaws.

Our adaptation algorithm is shown in Figure 3. The main loop is the standard plan refinement search loop. In addition to the pre-processing stage, we implement the following flaw types:

- **Open condition:** an event has a precondition not satisfied by any causal links from a temporally earlier event or the initial state.
- **Causal threat:** An event has an effect that undoes a condition necessary for another event to occur and there are no ordering constraints forbidding the interaction.
- **Un-decomposed event:** An abstract event has not been decomposed.
- **Dead end:** An event is not on a causal path to the outcome state.

Each flaw type is paired with one or more repair strategies. Repair strategies can be *additive* or *subtractive*.

Additive strategies are as follows. An open condition flaw can be repaired by instantiating a new event with an effect that unifies with the open precondition or by extending a causal link from an existing event to the open precondition (Penberthy and Weld 1992). Thus events are added to a plan in a backward-chaining fashion. A causal threat can be repaired by imposing ordering constraints between events (Penberthy and Weld 1992). An un-decomposed event can be repaired by selecting and applying a decomposition rule, resulting in new events instantiated, or existing events reused, as less abstract children of the abstract event (Young and Pollack, 1994).

Dead-end flaws can be handled in an additive fashion. We implement two additive dead-end repair strategies. First, if there is another event that has an open condition that unifies with an effect of the dead end, we can try to extend a causal link from an effect of the dead end to the open precondition of the other event. Second, we can shift an existing causal link to the dead-end event. This can happen if the dead end has an effect that matches the condition of a causal link between two other events. The dead-end event becomes the initiating point of the causal link, which may make the other event a dead end unless it has two or more causal links emanating from it. A third strategy is to ignore the flaw. This is used only as a last resort in the case that all other repair strategies, additive or subtractive, have proven to lead to failures. The intuition behind this strategy is that dead-end events are aesthetically undesirable but acceptable if necessary.

Subtractive strategies repair a flaw by deleting the source of the flaw from the plotline structure. Subtractive strategies are essential for plot adaptation because pre-existing events may interfere with the addition of new events, resulting in outright failure or awkward workarounds to achieve soundness and coherence. Deletion is straightforward. However, if an event to be deleted is part of a decomposition hierarchy, all siblings and children are deleted and the parent event is marked as un-decomposed. This preserves the intuition authored into quests and decomposition rules.

Open condition flaws can be subtractively repaired by deleting the event with the open precondition. Causal threat flaws can be subtractively repaired by deleting the event that threatens a causal link. Dead end flaws can be subtractively repaired by deleting the dead end event. We implement a heuristic that prefers to retain events in the original quests as much as possible.

The ability to add and delete events can lead to non-systematicity – the ability to revisit a node through different routes – and infinite loops. To preserve systematicity, we prevent the deletion of any event that was added by the algorithm. Events inserted by the algorithm are marked as "sticky" and cannot be subsequently deleted, whereas events in the original plotline are not sticky and can be removed.

Example

The short plotline in Figure 2 is meant to be played as an interactive role-playing game. In the game, the player kills the witch, gains the trust of the king, and is sent to rescue the princess, culminating in a wedding with the princess. However, suppose the player model predicts that the player

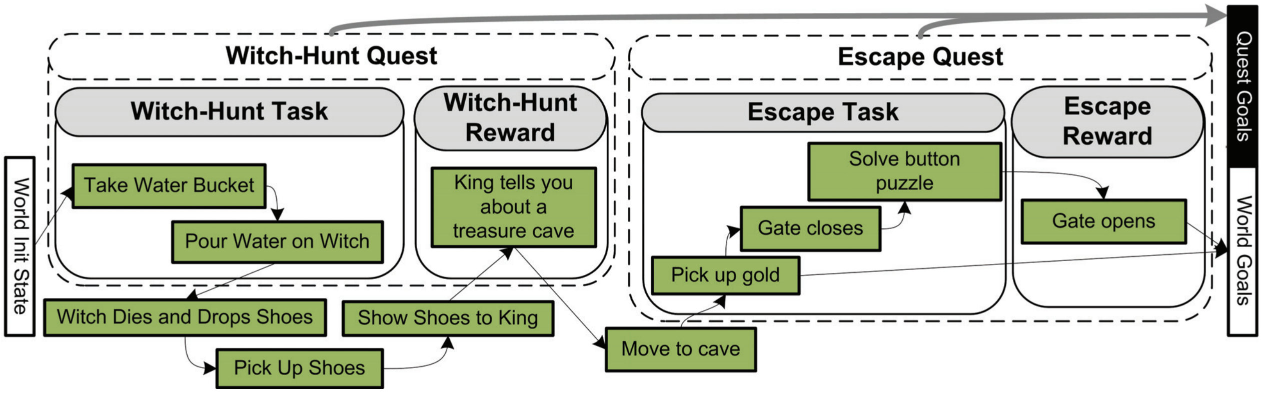

Figure 4. A complete and coherent narrative after adaptation

is not interested in rescuing and marrying a princess, but is instead motivated by acquisition of gold.

Based on input from the player model, the adaptor rewrites the outcome situation, removing `quest-complete(rescue)` and adding `quest-complete (escape)`. The `Rescue Quest` becomes a dead end and is deleted. To fulfill `quest-complete(escape)` in the outcome situation, the abstract event `Escape Quest` is instantiated. Decomposition rules are used to create child events until there are no more un-decomposed events. At this point, `Move to Palace` is a dead end and `Pick up Gold` (added during decomposition of `Escape Quest` – see Figure 4) has an open condition requiring the player to be at a cave. Turning to the open condition, the algorithm adds `Move to Cave` and then `King Tells about Cave` (initially not part of any hierarchy) to satisfy new open conditions. This eventually links to `Show Shoes`.

At this point the algorithm cannot progress any farther without deleting events. `Move to Palace` is still a dead end and there are no valid strategies except to delete it or ignore it. It is deleted and, over the next couple of iterations, `Move to Lair` and `Ask King about Lair` become dead ends and are also deleted.

This leaves `King Trusts You` as a dead end. Since it is part of an event hierarchy, when it is deleted the entire hierarchy is deleted (if it had siblings they would have been deleted, too) and `Witch-Hunt Quest` is marked as no longer completely decomposed. To decompose the quest, a decomposition rule is found that re-establishes the reward component of the quest by reusing the `King Tells about Cave` event, making the event a child of the quest. There are no more flaws. The final plotline is shown in Figure 4.

Authoring

Plotline adaptation scales up the ability to deliver customized experiences without significantly increasing the authoring effort. Chen et al. (2009) defines *authorial leverage* as the quality of experience per unit of domain engineering, where quality is a function of complexity, ease of change, and variability of experience. We focus on variability – the number of distinct stories – as our metric.

A one-time authoring cost by a domain engineer is incurred in the development of a world domain model, containing specifications for primitive events, abstract events (including quests), and decomposition rules. The payoff is a theoretically exponential leverage. The adaptation process can theoretically produce as many variations of a given plotline as the size of the power set of available quests. In practice, the number will be lower because a large fraction (e.g. 70%) of the original will be retained in each adaptation request. However, the scaling will still be exponential if the fraction remains constant. To manually achieve this scaling, one would have to author $n(n\text{-}1)$ transitions between quests (n-1 variations of each quest so it can be paired with n-1 other quests). Thus, one strength of plotline adaptation is the ability to opportunistically discover new transitions between quests based on the world model. Future work is required to measure the pragmatic authorial leverage of the system.

Evaluation

The principles of narrative soundness and coherence guide the adaptation process. To evaluate our approach to adaptation with respect to the necessity of detecting and resolving narrative soundness and coherence, we used an *ablative* technique whereby we determined degree of adaptation success on specific problems with several versions of the algorithm with different repair strategies disabled. Our hypothesis is that plotlines generated by the complete algorithm are preferred to stories generated when the system cannot repair dead ends or open preconditions.

Two adaptation tasks were performed based on a hypothetical player model. Each required the replacement of one quest with another in a two-quest plotline. The following versions of our algorithm were used to generate three versions of plotlines for each task:

- **N0:** Cannot repair flaws except un-decomposed events
- **N1:** Cannot repair dead-end flaws
- **N2:** The complete algorithm

Plotlines produced by N0 lacked events that establish required preconditions and seemed to contain gaps. Plotlines produced by N1 contained at least one dead end. Text descriptions of each plotline were hand-authored and participants were provided with the six descriptions arranged in two groups where each group contained

Table 1. Results of the second study

	N2>N1	N2>N0	N1>N0
Plot group 1	52%	88%*	88%*
Plot group 2	76%*	100%*	60%

adaptations generated by N0, N1, and N2 for one of the two tasks. Our hypothesis is confirmed if people prefer N2 to N1 (N2>N1) and N2 to N0 (N2>N0).

Twenty-five participants were involved in the study. The results are summarized in Table 1. All results were put to one-sided tests on binomial distribution at the significance level of $p < 0.05$; asterisks (*) mark significant results. In group 1, a significant number of participants preferred N2 to N0, but no significance was found about those who preferred N2 to N1. For plotlines in group 2, a significant number of participants preferred N2 to both N0 and N1.

Results from group 1 and group 2 should corroborate, suggesting a hidden independent variable. The N1 plotlines in both groups contained a dead end. However, the group 1 dead end appeared to be events that were never followed up, whereas the group 2 dead end directly contradicted the apparent intentions of other events. It is likely that our system, using formal definitions, is more sensitive to story incoherence than human game players. Thus, we believe that group 2 plotlines, consisting of more disruptive and noticeable dead ends, are more representative of worst-case situations. Group 2 results indicate that it may be beneficial to be cautious, erring on the side of being overly sensitive to story incoherence. Results of Group 2 validate our hypothesis, leading us to believe that enforcing narrative coherence is beneficial and that no harm is done by being overly sensitive to story incoherence.

Conclusions

As game players possess different motivations, tastes and needs, a one-size-fits-all approach to game plotlines may prove to be limiting. We treat adaptation as the optimization of plotlines based on requirements derived from a player model employing knowledge about player preferences and a model of novelty. As such, we find an offline approach to be beneficial in achieving global optimization of plotline structure.

The adaptation problem itself is solved by an iterative improvement search based on partial order planning. However, in order to start from a complete plotline and arrive at a variation with different quests, we employ both additive and subtractive improvement mechanisms. To the extent that the player model is an approximation of player preferences, future work may pair our offline adaptation technique with online interactive storytelling engines.

As the world orients toward greater on-demand and customized entertainment experiences, overcoming the content authoring bottleneck will increasingly require automation on the level of creative production. We believe that a partnership between human authors and automated adaptation can scale up our ability to deliver the “right story to the right person at the right time.”

References

Chen, S., Nelson, M. and Mateas, M. 2009. Evaluating the Authorial Leverage of Drama Management. In *Proc. of the 5th AI and Interactive Digital Entertainment Conf.*

Lebowitz, M. 1987. Planning stories. In *Proceedings of the 9th Annual Conference of the Cognitive Science Society*.

Muñoz-Avila, H. and Cox, M. 2008. Case-Based Plan Adaptation: An Analysis and Review. *IEEE Intelligent Systems* 23(4).

Penberthy, J. and Weld, D. 1992. UCPOP: A Sound, Complete, Partial-Order Planner for ADL. In *Proc. 3rd Int’l. Conf. on Knowledge Representation and Reasoning*.

Porteous, P. and Cavazza, M. 2009. Controlling Narrative Generation with Planning Trajectories: The Role of Constraints. In *Proc. of the 2nd Int’l. Conf. on Interactive Digital Storytelling*.

Riedl, M.O., Stern, A., Dini, D., and Alderman, J. 2008. Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training. *International Transactions on System Science and Applications* 3(1).

Riedl, M.O. and Young, R.M. In press. Narrative Planning: Balancing Plot and Character. *Journal of AI Research*.

Roberts, D.L and Isbell, C.L. 2008. A Survey and Qualitative Analysis of Recent Advances in Drama Management. *International Transactions on Systems Science and Applications* 3(1).

Rollings, A. and Adams, E. 2003. *Andrew Rollings and Ernest Adams on Game Design*. New Riders.

Saunders, R. and Gero, J. 2004. Curious Agents and Situated Design Evaluations. *AI for Engineering, Design, Analysis, and Manufacturing* 18(2): 153-161.

Seif El-Nasr, M. 2007. Interaction, Narrative, and Drama Creating an Adaptive Interactive Narrative using Performance Arts Theories. *Interaction Studies* 8(2).

Sharma, M., Ontañón, S. Mehta, M., and Ram, A. 2010. Drama Management and Player Modeling for Interactive Fiction Games. *Computational Intelligence* 26(2):183-211.

Thue, D., Bulitko, V., Spetch, M., and Wasylishen, E. 2007. Interactive storytelling: A player modeling approach *Proc. 3rd AI and Interactive Digital Entertainment Conf.*

Trabasso, T. and van den Broek, P. 1985. Causal Thinking and the Representation of Narrative Events. *Journal of Memory and Language* 24: 612-630.

Yee, N. 2006. The Demographics, Motivations, and Derived Experiences of Users of Massively-Multiuser Online Graphical Environments. *PRESENCE: Teleoperators and Virtual Environments* 15(3): 309-329.

Young, R.M. 1999. Notes on the Use of Plan Structures in the Creation of Interactive Plot. *Working Notes of the AAAI Fall Symposium on Narrative Intelligence*.

Young, R. M. and Pollack, M.E 1994. Decomposition and causality in partial-order planning. In *Proc. of the 2nd Int’l. Conf. on AI and Planning Systems*.