

Creating Customized Game Experiences by Leveraging Human Creative Effort: A Planning Approach

Boyang Li and Mark O. Riedl

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

Abstract. The task of entertaining people has, until very recently, been the exclusive domain of humans. However, recent advances in Artificial Intelligence (AI) suggest that intelligent systems may be used to create dynamic and engaging real-time entertainment experiences. In this paper we consider a novel technique called *Experience Adaptation*. Experience Adaptation is an offline process that leverages human creative ability by taking human-authored specifications of desired user experiences and autonomously “re-writing” them based on unique requirements of individual users. In this chapter, we illustrate Experience Adaptation in the context of computer-based role-playing games in which player experience is highly dependent on an unfolding plotline. Our approach uses a 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.

Keywords: Experience Adaptation, Narrative Intelligence.

1 Introduction

Artificial intelligence has long been used to automate certain tasks in order to perform those tasks faster, more accurately, more efficiently, more safely, or more often. However, the task of entertaining people has, until very recently, been the exclusive domain of humans. When it comes to commercial production of entertainment artifacts like TV shows, movies, novels, theatre, computer games, etc., the task of entertaining people has been the exclusive domain of “creative professionals” such as writers, actors, movie directors, theatre and improv performers, dungeon masters, and so on. The reason the task of entertaining people has been the exclusive domain of humans is that the creativity and intuition that human entertainers possess have not been reliably replicated in computational systems.

Currently, there are fewer professional and expert human “producers” of entertainment than there are human “consumers” of entertainment. This model works fine for mass-consumption entertainment such as film, TV, books, and, to a lesser extent, theatre performances. The *creative authoring bottleneck* refers to the situation where the cost of employing enough professional human producers to satisfy the demands of human consumers is prohibitively high, resulting in a situation where

there is more demand for quality content than production of quality content. (We use “authoring” to refer to the deliberate creation of any entertainment-related artifact, including an improvised performance created in real-time [9]). Recent work in the area of computational creativity, story generation, interactive storytelling, and autonomous believable agents lays the groundwork for a future where entertainment is fully automated. We are now at a unique point where modern computer technology, simulation, and computer games have opened up the possibility of that more can be done in the area of *on-demand* and *uniquely customized* entertainment.

- **On-demand entertainment** refers to the possibility that one can request, at any time, an entertainment experience that is significantly different from any previously consumed. For example, game players can exhaust game-play content faster than expansion packs and new releases can be produced. For an early case study in which consumers outpace producers of content in online virtual game worlds, see [11]. Ideally, there is a one-to-one relationship between producers and consumers so that content can never be consumed faster than it is produced.
- **Uniquely customized entertainment** means that entertainment artifacts should be customized or configured to suit every player’s unique motivation, tastes, desires and history. Usually this information is not available at the time the game is designed and implemented. The customization decisions can only be made in a just-in-time fashion because we need to know (a) who the user is, (b) what the user’s motivation, tastes, and desires are, and (c) what the user is doing at any given moment.

As we approach a world in which on-demand and uniquely customized entertainment is the expectation, the conventional consumer-producer model breaks down. To overcome the creative authoring bottleneck, we must consider automation. In this chapter we consider a technique called *Experience Adaptation* [8, 17]. Experience Adaptation is an offline process that leverages human creative ability by taking human-authored specifications of desired future experiences and autonomously “re-writing” them based on unique requirements of individual users.

To motivate the need for on-demand and uniquely customized entertainment, we explore these concepts in the context of generating plotlines for computer-based role-playing games. Computer based role-playing are believed to be highly dependent on individual differences such as play styles [1, 29, 26] and involve numerous tasks that may or may not be of interest to players. Rollings and Adams [21] argue that the core of gameplay in any game is “one or more causally linked series of challenges in a simulated environment.” These challenges often appear in units of role-playing game storytelling called quests. To accomplish the quests, players have to perform required gaming activities such as combat or puzzle-solving in a virtual world. Game designers usually use a main plotline, comprised of a set of quests, often ordered, that are sufficient and necessary to complete the game. The main plotline provides the player with a sense of meaningful progression through the game. Although the main plotline is mandatory, optional side-quests are often available to augment the gameplay experience, and to afford players a limited degree of customization through choice. Instead of supplementary side quests, we investigate intelligent systems that adapt and customize the primary plotline to satisfy player preferences, needs, and desires while maintaining narrative coherence and preserving the original plotline author’s intent.

We argue that customization of entertainment experience involves presenting *the right experience to the right person at the right time*. The significance of this claim is twofold. First, players usually possess diverse motivation, tastes, desires and history. A one-size-fits-all script may not cater to all types of players. Moreover, to achieve optimal game experience, challenges must adapt to the player's skill level. Secondly, preferences of players can change over time. Having experienced 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.

The remainder of the chapter is organized as follows. In Section 2, we formulate the problem of game experience adaptation and ground the notion of experience on discrete computational representations of narrative. In Section 3, we provide a mathematical notion of narrative coherence based on our representation. Section 4 deals with the practical side of experience adaptation with a detailed planning algorithm, an example, and discussions of authoring and evaluation. Section 5 provides discussion of related work.

2 Experience Adaptation

We believe a computational system that scales up a human creator's ability to deliver customized experiences to a large number of consumers will provide a solution to the content creation bottleneck. Automated adaptation of experience is necessary when we can only learn about our intended customer at playtime. Unfortunately, the construction of autonomous systems capable of assuming responsibility for human users' entertainment experiences is largely an open research question. Until we have computational systems capable of creativity rivaling that of human experts, there is value in exploring *hybrid* approaches in which humans and computational systems share the responsibility of managing human users' entertainment experiences. Thus, such a computational system becomes a practical compromise: it should be able to facilitate human authors and scale up their authoring effort, so that a large number of customized variations of the original content can be produced easily without sacrificing the quality. Chen et al. [2] coined the term *authorial leverage* to indicate the ratio of quality of experience delivered by a computational system to authorial input. Hybrid Experience Adaptation systems leverage human knowledge for the purpose of creating novel experiences.

2.1 Leveraging Human Creative Effort

In the context of computer games, Experience Adaptation takes a few human-authored descriptions of experiences to be had in a virtual world and provides numerous experiences customized to individuals. The Experience Adaptation pipeline is shown in Fig. 1. A human author develops a plotline as a means of describing what a user should experience in the virtual world. The storyline determines events that will happen in the virtual world, including specifications for the behaviors of non-player characters. The plotline, provided in a computational format that facilitates

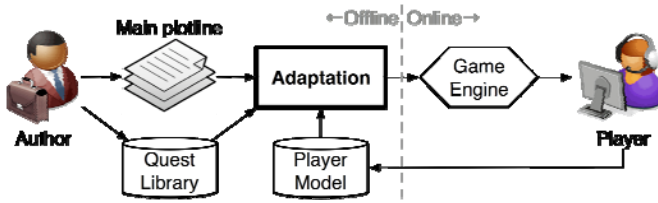


Fig. 1. The plotline adaptation architecture

automated analysis and reasoning, is combined with a player model and a world model. The world model describes what characters – human or virtual characters – can do in the world, and how the world is changed when actions are performed. The player model provides information about the user in terms of preferences over experiences. The player model also contains historical information describing the types of experiences the user has previously had. The player model is capable of generating a set of experiential requirements – the features of the experience the user should receive.

The plotline, player model requirements, and world model are inputs into the Experience Adaptor. The plotline is analyzed to determine whether it meets the experiential requirements from the player model. If it does not, the Experience Adaptor engages in an iterative process of making changes to the plotline until it meets the requirements of the user model. The result is a new creative artifact describing a customized narrative experience, which is sent to a game engine for interactive real-time execution. Note the cycle in Fig. 1 created by the Experience Adaptation process, resulting in improved replayability of authored experiences; as the player model evolves over time, the same human-authored storyline can be recycled into novel experiences.

The core component in the Experience Adaptation process is the Experience Adaptor. The Experience Adaptor has two functions, to interpret the requirements provided by the user model, and to “rewrite” the story provided by a human author. The Experience Adaptation Problem is as follows: given a domain model, a set of experiential requirements, and a storyline that does not meet the requirements, find a coherent storyline that meets the experiential requirements and preserves the maximal amount of original content. A coherent storyline is one in which all events have causal relevance to the outcomes [28]. The preservation of original content ensures that as much of the creative intuition of the human author remains intact as possible.

The plotline can be adapted in three different ways:

- **Deletion:** Events in the storyline can be removed because they are unnecessary or unwanted.
- **Addition:** Events can be added to the storyline to achieve experiential requirements, and to ensure narrative coherence.
- **Replacement:** a combination of deletion and addition, old events are swapped for new events that better achieve experiential requirements.

The application of these operations enables a refinement-search algorithm to incrementally tear down and build up a complete, human-authored narrative structure

until it meets the experiential requirements. Experience Adaptation can be online or offline; we have chosen to implement an architecture with an offline Experience Adaptor so as to optimize the overall global structure of the experience.

For the Execution Adaptor to function, it needs a set of experiential requirements that it can use to evaluate the current plotline and evaluate potential new plotlines. The player model is responsible for generating this set of experiential requirements. 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 by Sharma et al. [25] 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 work by Saunders and Gero [23] 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.

2.2 Computational Representation of Plot

Experience Adaptation can only work if experience can be formally represented in a form that can be reasoned about and manipulated. As noted above, experiences are captured as narratives. Following others [30, 18, 15], 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. The plan representation provides a formal framework to explicitly represent causal relationships between events and reason about them on first principles (for example, we can ask if a narrative is sound). Further, plans closely resemble cognitive models of narrative. Graesser et al. [4] and Trabasso and van den Broek [28] in particular highlight the importance of causalities in stories. 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 [14]. 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. The set of decomposition rules act as a grammar 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.

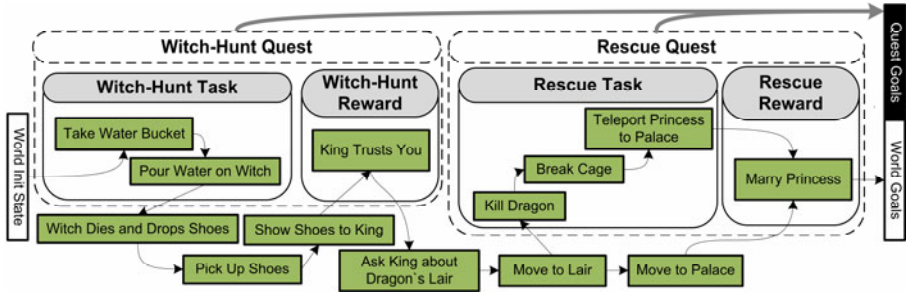


Fig. 2. An original game plotline before adaptation. The plotline contains two quest structures, represented as hierarchical decompositions.

In our system, quests are represented as top-level abstract events. A quest has a single effect, $\text{quest-complete}(\text{quest-X})$, 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. Fig. 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 from the king. Arrows represent causal links. Note that not all causal links are shown for clarity's sake. Temporal links are omitted.

The quest library (see Fig. 1) is a model of the dynamics of the virtual world. It is made up of primitive and abstract event templates plus decomposition rules. Event templates are parameterized events represented in a STRIPS-like (cf., [3]) format, allowing for specific characters, props, and location to be substituted in when an event is instantiated into a plan. The main plotline of the game, an example of which is illustrated in Fig. 2, is comprised of instantiated events, causal links, temporal links, and event relationships.

3 Narrative Coherence

We believe partial-order plans are effective representations of stories. Thus, a reasonable approach to solving the Experience Adaptation Problem is to use a form of refinement search that can manipulate partial-order plans. However, conventional planning is geared towards maximum efficiency, whereas the shortest or most efficient sequence of actions is rarely the best or most coherent story. Therefore, special care must be taken to maintain the coherence of the story generated.

Trabasso and van den Broek [28] proposed the idea of *narrative coherence* as a property of the causal structure of the story. A narrative is coherent when each event contributes significantly to the causal achievement of the main outcome. On each hierarchical level, a plan can be seen as a directed acyclic graph (DAG) with actions represented as vertices and causal links as edges. Whereas soundness is achieved if all

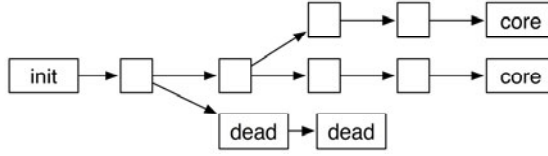


Fig. 3. Schematic of dead-end events

preconditions are on causal chains back to the initial state without creating logical inconsistencies, narrative coherence is achieved when each event has at least one effect on a causal chain to the outcome state. In this section, we elaborate on two types of story flaws that break narrative coherence: *dead ends* and *superfluous efforts* [7]. These flaws can happen even in a sound plan. The definitions of the two flaws rely purely on the abstract causal structure and performers of actions. In other words, the flaws are defined independently of the story domain, although they are dependent on how the preconditions and effects of actions are defined.

3.1 Core Set

First, we suggest that some events in a story are of special interest to the audience and more important than others. The significance of events can be perceived by human designers and audience. Other events set context for, revolve about, and eventually lead to these events, which form the core set of the story. The core set depends on the application. For example, when the player is interested in becoming filthy rich, the event where treasures are obtained is crucial, and other events should be subordinate. In this paper, we define the core set to include only the goal state of the plan. However, depending on the circumstances, one may want to choose other events for the core set. For example, complex authorial intent may be represented in the plan as intermediate goals, which can be negated after being achieved [16].

3.2 Dead Ends

An event is a dead end if it does not contribute in a meaningful way to the unfolding of events in the core set. It is believed that the presence of dead-end events directly harms the perception of narrative coherence. Following the previous example, suppose the primary interest of the player is to find treasure, then the event of obtaining a sword which is not useful for this purpose is not very relevant. Therefore, we consider the event to be a dead end. See Fig. 3 for an illustration of the causal structure of dead end, where a box represents an action and an arrow represents a causal link. The initial state, core events, and dead ends are labeled.

Formally, in a story DAG $G = (V, E)$ where a vertex $v \in V$ represents events in the plan and $(u, v) \in E$ if and only if any effect of event u satisfies at least one precondition of event v . We use $\text{path}(u, v)$ to denote the fact that there is a path from vertex u to vertex v in G . Given a core set $S_C \subseteq V$, the set of dead end actions S_D is defined by:

$$\forall u \in V, v \in S_C, \neg \text{path}(u, v) \Leftrightarrow u \in S_D. \quad (1)$$

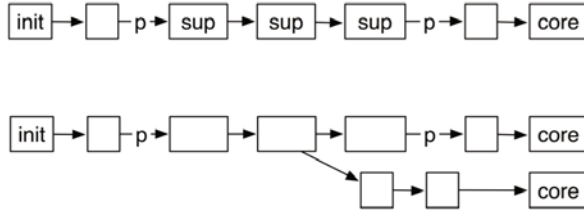


Fig. 4. Schematic of superfluous efforts (top) and non-superfluous efforts (bottom)

In general, it is recommended the core set be designed such that there are no dead ends in the original hand-authored storyline.

3.3 Superfluous Effort

Another breach of narrative coherence could happen when an event is part of a causal chain that contributes to the core set but at closer inspection appears to reestablish a world condition that is unnecessarily negated. For example, the player gives a sword to a stranger, and then has to steal it back to slay a dragon with it. The action of giving the sword is superfluous if, before the condition of the player having the sword, no other effects contribute to the core set. Fig. 4 (top) shows superfluous effort because the events serve no purpose other than re-establishing condition p . Fig. 4 (bottom) shows non-superfluous effort because the events that re-establish condition p serve an additional purpose. It is required that actions in the superfluous efforts are all performed by the same character.

Formally, a subset of vertices $S \subseteq V$ is a superfluous effort if:

1. S is (weakly) connected.
2. The set of conditions annotating outgoing edges is a subset of the set of conditions annotating incoming edges.
3. $\neg \exists a \in V, (\exists b, c \in S, \text{path}(b, a) \wedge \text{path}(a, c))$

Whereas, dead ends prevent interference with intentions of the author, superfluous efforts can be considered a heuristic guard against interference with intentions of story characters. The list of coherence flaws is by no means exhaustive, but the two examples illustrate two very important and complimentary aspects of the narrative coherence. We believe that the preservation of narrative coherence is important for any type of story adaptation.

4 The Experience Adaptor

The Experience Adaptor is the central component of the Experience Adaptation process (see Fig. 1). It leverages existing plotlines and promotes replayability by creating a cycle of play and adaptation. The Experience Adaptor 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 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.

4.1 Experience Adaptation Planning Algorithm

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 [14] 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 Fig 5. 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.
- **Superfluous effort:** Events reestablish a redundant world 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 [14]. Thus

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

function ADAPT (*plan*, *requirements*, A) **returns** solution or failure
 $plan \leftarrow \text{REWRITE-GOAL-AND-INITIS}(plan, requirements)$
 $fringe \leftarrow \{plan\}$
 loop do
 if $fringe = \emptyset$ **then return failure**
 $plan \leftarrow \text{POP}(fringe)$
 if $plan$ has no flaws **then return plan**
 $flaw \leftarrow \text{GET-ONE-FLAW}(plan)$
 $newplans \leftarrow \text{REPAIR}(flaw, plan, A)$
 $fringe \leftarrow \text{INSERT-AND-SORT}(newplans, fringe)$

Fig. 5. The plotline adaptation algorithm

events are added to a plan in a backward-chaining fashion. A causal threat can be repaired by imposing ordering constraints between events [14]. 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 [31].

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.

Superfluous effort flaws often occur when resolving other flaws. To repair a superfluous effort, one strategy is to extend causal links from events in the superfluous effort back to earlier events with effects that match. Extending causal links back to earlier events is a common technique used in continuous planning [22]. After the extension, some events in the superfluous effort become dead ends, and will be repaired accordingly. As with dead ends, a last-resort strategy is to ignore the flaw, favoring a narrative with superfluous efforts over no solution.

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. Table 1 shows all the repair strategies available for each type of flaw.

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 or link that was added by the algorithm. Events and links inserted by the algorithm are marked as “sticky” and cannot be subsequently deleted, whereas those in the original plotline are not sticky and can be removed.

4.2 Heuristics

As with all search problems, a powerful heuristic can significantly improve the efficiency of the search algorithm. Two types of heuristics are typically used in conventional partial-order planning. Here, we focus on the heuristic that determines which plan on the fringe to visit. Traditionally, such a heuristic favors plans with fewer flaws and shorter plans over longer ones.

Table 1. Additive and subtractive strategies for repairing flaws

Flaw	Description	Repair Strategies
Open condition	Event e has a precondition p that is not satisfied by a causal link.	<ol style="list-style-type: none"> 1. Instantiate new event e_{new} that has an effect that unifies with p. Extend a causal link from e_{new} to e. 2. Select an existing event e_{old} that has an effect that unifies with p. Extend a causal link from e_{old} to e. 3. Delete e.
Causal threat	Event e_k has an effect that negates a causal link between events e_i and e_j .	<ol style="list-style-type: none"> 1. Promotion: temporally order e_k before e_i. 2. Demotion: temporally order e_k after e_j. 3. Delete e_k.
Un-decomposed event	Event e is abstract but has no children.	<ol style="list-style-type: none"> 1. Select and apply a decomposition rule, instantiating new events or reusing existing events as children.
Dead end event	Event e is a dead end.	<ol style="list-style-type: none"> 1. Select an existing event e_{old} that has a precondition that is unsatisfied and that unifies with an effect of e. Extend a causal link from e to e_{old}. 2. Select an existing event e_{old} that has a precondition that is satisfied by causal link c and unifies with an effect of e. Transfer the starting point of c to e. 3. Instantiate new event e_{new} that has a precondition that unifies with an effect of e. Extend a causal link from e to e_{new}. 4. Ignore the flaw.
Superfluous event	Event e is superfluous.	<ol style="list-style-type: none"> 1. Link effects of earlier steps to preconditions of e. 2. Ignore the flaw.

In order to preserve the original authorial intent of the plotline, deletion of events should be used with caution and guided by a good heuristic. One method is to favor the deletion of actions more relevant to the quests removed than to quests that remain. We propose two relevance criteria to build such a heuristic. The first criterion of relevance is causal relationships. Actions that immediately precede or follow actions within removed quest decompositions are more relevant to them than actions further away. The causal relevance between two actions is inversely proportionate to the length of the shortest path between them. The second criterion is the objects or characters the actions refer to. For example, actions in the *Witch-Hunt* quest refers to the witch frequently, where as other quests, as shown in Fig. 2 and 8, do not refer to her at all. We propose that the locality of character and object reference can be exploited to identify relationships between events in a plan.

4.3 Adaptation Example

In this section, we explain the working of quest-centric adaptation planning with an example of a simple role-playing game. As shown in Figure 6, the original game narrative consists of two quests. In the first quest, the player kills the witch, arch-enemy of the king, by pouring a bucket of water on her. In the second quest, the player rescues the princess from a dragon and marries her. However, suppose the player prefers treasures to marriage, we can remove the rescue quest and add an escape quest where the player is locked in a treasure cave and can only escape by solving a puzzle. The original storyline, an intermediate step, and the final result are shown respectively in Figs. 6, 7 and 8. The order of operations is denoted with numbers in circles. We do not intend to explain every detail due to space constraints. For the sake of simplicity, the search is assumed to be nondeterministic, which always makes the correct choice at every decision point. Backtracking will happen in real applications, even though not shown here.

Fig. 6 shows a given storyline of two quests. Thick gray arrows indicate the two quests satisfy quest-level goals `quest-complete(Witch-Hunt)` and `quest-complete(Rescue)` respectively. At the world-state level, the only goal is `married(player, princess)`, which is satisfied by the primitive action `Marry Princess` as shown by a thin black arrow.

We begin with requirements from the user preferring escape missions to rescues. The quest-level goal situation is updated accordingly by removing `quest-complete(rescue)` and adding `quest-complete(escape)`. The only outgoing causal link from the action `Rescue Quest` is used to satisfy this quest-level goal. As a result, this action becomes a dead end. The first step of planning is to remove it together with all descendant actions and all associated causal links. To fulfill the added goal `quest-complete(escape)`, the abstract action `Rescue Quest` is added and subsequently decomposed. New actions in the decomposition are added. They bring new open preconditions. We then deal with world-level goals. In the next few refinement iterations, dead ends, marked with number 3, are removed and actions marked with number 4 and 5 are added to fulfilling open preconditions. After these operations, we have obtained the plan in Fig. 7.

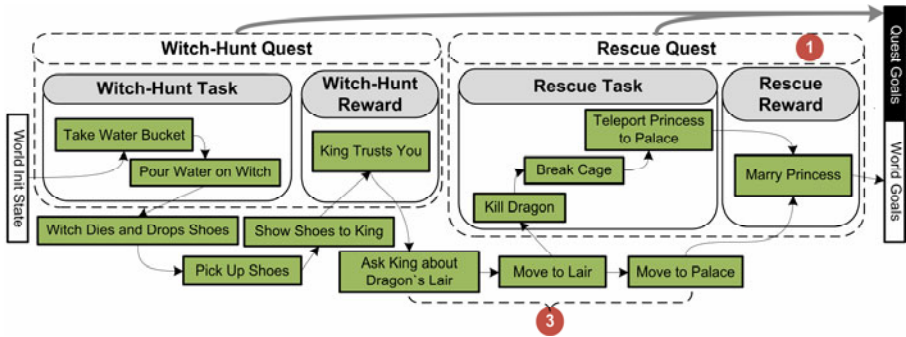


Fig. 6. The original plotline

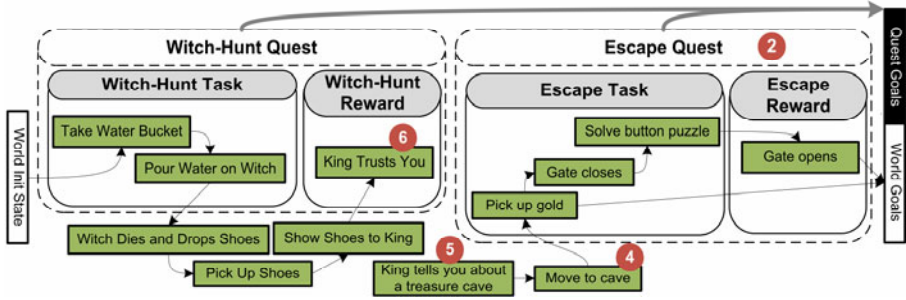


Fig. 7. Snapshot of an intermediate point in the adaptation process

The reward component of Witch-Hunt Quest is modified as follows. The action King Trusts You, marked with 6, becomes a dead end and removed. Its removal introduces two flaws: 1) the action Show Shoes to King has become a dead end, and 2) the Witch-Hunt Reward abstract action now has no decomposition. The relevance heuristic comes into play in resolving the dead end. The action Show Shoes to King is determined to be more relevant to the remaining quest than to the removed. Hence, we prefer establishing an outgoing link known-success(king, hero, witch-hunt) for action number 5 to removing it. Finally, we need a new decomposition for Witch-Hunt Reward, and we realize the decomposition can reuse action number 5. Having fixed all flaws, we have a complete and coherent narrative, shown in Fig. 8.

4.4 Analysis of Authorial Leverage

Plotline adaptation scales up the ability to deliver customized experiences without significantly increasing the authoring effort. Chen et al. [2] 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.

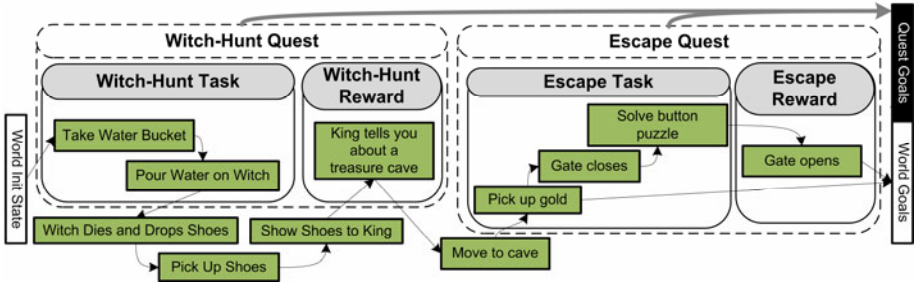


Fig. 8. Complete, Coherent Narrative after Adaptation

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 $n(n-1)$ 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.

4.5 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 2. Empirical results of the evaluation

Plot Group 1	N2>N1	N2>N0	N1>N0
No. Participants	13	22	22
Percentage	52%	88%*	88%*
Plot Group 2	N2>N1	N2>N0	N1>N0
No. Participants	19	25	15
Percentage	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 2. 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.

5 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 [20] 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. [26] describe a technique whereby a player model based on role player types is used to select branches through an interactive story. Seif El-Nasr [24] attempts to infer feature-vectors representing player style, affecting changes in which dramatic content is presented to the player. Sharma et al. [25] 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 [18], a planning-based interactive narrative system.

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., [10], [6], [15], and [19]). 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 [12]. However, in the average case, starting from an existing plotline will require much fewer decisions to be made.

In a parallel effort, the TACL system [13] is designed to adapt and customize military training scenarios. Realistic military training is a highly rigorous process. Any automatic adaptation must preserve pedagogical correctness and the tolerance of modification is low. Game quests, on the other hand, can be modified extensively. In this paper, we apply the algorithm in the novel context of quests and games.

Work on adapting player experience in games has been addressed in terms of game level generation. Hullett and Mateas [5] have investigated generation of game level floor plans, and thus the narrative of moving through space, using HTN planning. HTN planning requires complete specification of how each task can be performed. In comparison, our approach is capable of opportunistic discovery of novel event sequences. Finally, others have explored game world generation and other non-narrative content generation using neural network models of players and evolutionary computation (cf., [27]). At the moment, we are ignoring the generation of landscape and environment in games.

6 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 experience to the right person at the right time.”

References

1. Bartle, R.: Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs. *Journal of Virtual Environments* 1 (1996)
2. Chen, S., Nelson, M.J., Sullivan, A., Mateas, M.: Evaluating the Authorial Leverage of Drama Management. In: Mehta, M., Louchart, S., Roberts, D.L. (eds.) *Intelligent Narrative Technologies II: Papers from the Spring Symposium* (Technical Report SS-09-06). AAAI Press, Menlo Park (2009)
3. Fikes, R., Nilsson, N.: STRIPS: a New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence* 2, 189–208 (1971)
4. Graesser, A., Lang, K.L., Roberts, R.M.: Question Answering in the Context of Stories. *Journal of Experimental Psychology: General* 120, 254–277 (1991)
5. Hullett, K., Mateas, M.: Scenario Generation for Emergency Rescue Training Games. In: *Proceedings of the Fourth International Conference on the Foundations of Digital Games* (2009)
6. Lebowitz, M.: Planning Stories. In: *Proceedings of the 9th Annual Conference of the Cognitive Science Society* (1987)
7. Li, B., Riedl, M.O.: Planning for Individualized Experiences with Quest-Centric Game Adaptation. In: *Proceedings of the ICAPS 2010 Workshop on Planning in Games* (2010)
8. Li, B., Riedl, M.O.: An Offline Planning Approach to Game Plotline Adaptation. In: *Proceedings of the 6th Conference on Artificial Intelligence for Interactive Digital Entertainment* (2010)
9. Magerko, B.S., Riedl, M.O.: What Happens Next?: Toward an Empirical Investigation of Improvisational Theatre. In: *Proceedings of the 5th International Joint Workshop on Computational Creativity* (2008)
10. Meehan, J.: *The Metanovel: Writing Stories by Computer*. Ph.D. Dissertation. Yale University, New Haven (1976)
11. Morningstar, C., Farmer, F.R.: The Lessons of Lucasfilm’s Habitat. In: Bendikt, M. (ed.) *Cyberspace: First Steps*. MIT Press, Cambridge (1991)
12. Muñoz-Avila, H., Cox, M.: Case-Based Plan Adaptation: An Analysis and Review. *IEEE Intelligent Systems* 23 (2008)
13. Niehaus, J., Riedl, M.O.: Scenario Adaptation: An Approach to Customizing Computer-Based Training Games and Simulations. In: *Proceedings of the AIED 2009 Workshop on Intelligent Educational Games* (2009)

14. Penberthy, J., Weld, D.: UCPOP: A Sound, Complete, Partial-Order Planner for ADL. In: *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning* (1992)
15. Porteous, P., Cavazza, M.: Controlling Narrative Generation with Planning Trajectories: The Role of Constraints. In: Iurgel, I.A., Zagalo, N., Petta, P. (eds.) *ICIDS 2009*. LNCS, vol. 5915, pp. 234–245. Springer, Heidelberg (2009)
16. Riedl, M.O.: Incorporating Authorial Intent into Generative Narrative Systems. In: Mehta, M., Louchart, S., Roberts, D.L. (eds.) *Intelligent Narrative Technologies II: Papers from the Spring Symposium (Technical Report SS-09-06)*. AAAI Press, Menlo Park (2009)
17. Riedl, M.O., Li, B.: Creating Customized Virtual Experiences by Leveraging Human Creative Effort: A Desideratum. In: *Proceedings of the AAMAS 2010 Workshop on Collaborative Human/AI Control for Interactive Experiences* (2010)
18. Riedl, M.O., Stern, A., Dini, D., Alderman, J.: Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training. *International Transactions on Systems Science and Applications* 3 (2008)
19. Riedl, M.O., Young, R.M.: Narrative Planning: Balancing Plot and Character. *Journal of Artificial Intelligence Research* 39 (2010)
20. Roberts, D.L., Isbell, C.L.: A Survey and Qualitative Analysis of Recent Advances in Drama Management. *International Transactions on Systems Science and Applications* 3, 61–75 (2008)
21. Rollings, A., Adams, E.: *Andrew Rollings and Ernest Adams on Game Design*. New Riders, Indianapolis (2003)
22. Russell, S., Norvig, P.: *Artificial Intelligence: A modern approach*. Prentice-Hall, Englewood Cliffs (2002)
23. Saunders, R., Gero, J.: Curious Agents and Situated Design Evaluations. *AI for Engineering, Design, Analysis, and Manufacturing* 18, 153–161 (2004)
24. Seif El-Nasr, M.: Interaction, Narrative, and Drama Creating an Adaptive Interactive Narrative using Performance Arts Theories. *Interaction Studies* 8 (2007)
25. Sharma, M., Ontañón, S., Mehta, M., Ram, A.: Drama Management and Player Modeling for Interactive Fiction Games. *Computational Intelligence* 26, 183–211 (2010)
26. Thue, D., Bulitko, V., Spetch, M., Wasylishen, E.: Interactive Storytelling: A Player Modelling Approach. In: *Proceedings of the 3rd Conference on Artificial Intelligence and Interactive Digital Entertainment Conference* (2007)
27. Togelius, J., Yannakakis, G., Stanley, K., Browne, C.: Search-based Procedural Content Generation. In: *Proceedings of the EvoStar Conference* (2010)
28. Trabasso, T., van den Broek, P.: Causal Thinking and the Representation of Narrative Events. *Journal of Memory and Language* 24, 612–630 (1985)
29. Yee, N.: The Demographics, Motivations, and Derived Experiences of Users of Massively-Multiuser Online Graphical Environments. *PRESENCE: Teleoperators and Virtual Environments* 15, 309–329 (2006)
30. Young, R.M.: Notes on the Use of Plan Structures in the Creation of Interactive Plot. In: Mateas, M., Singers, P. (eds.) *Narrative Intelligence: Papers from the Fall Symposium (Technical Report FS-99-01)*. AAAI Press, Menlo Park (1999)
31. Young, R.M., Pollack, M.E.: Decomposition and causality in partial-order planning. In: *Proceedings of the 2nd International Conference on Artificial Intelligence and Planning Systems* (1994)