Planning for Individualized Experiences with Quest-Centric Game Adaptation

Boyang Li and Mark O. Riedl

School of Interactive Computing
Georgia Institute of Technology, Atlanta, GA, USA
boyangli@gatech.edu, riedl@gatech.edu

Abstract

Planning has been extensively used to build competent opponents in games for human players. In this paper, we focus not on winning strategies but on creating an enjoyable overall gaming experience by adapting human-authored game narratives and customizing them to the players' motivation, tastes and needs. We discuss the benefits of modeling game narratives as plans and analyze causal structures to build novel computational models of narrative coherence. A planning approach to the narrative adaptation problem is presented. The planner takes a complete storyline comprised of several quests and iteratively searches for modifications, deleting and inserting quests and events, until it meets the user's preferences. A user study strongly suggested the proposed notion of narrative coherence has positive influences on story aesthetics.

Introduction

Much research efforts on planning in games have been devoted to creating competent opponents that win as much as possible for human players. Various planning techniques have been applied in the context of real-time and turn-based strategy games (Chung, Buro, and Schaeffer 2005; Sánchez-Ruiz et al. 2007; Balla and Fern 2009), first-person shooters (Orkin 200), and contract bridge (Smith, Nau, and Throop 1998), to name just a few. These works focused on planning strategies for an AI player, and have achieved success to some extent.

However, relatively few works have focused on optimizing gaming experience of players. As pointed out by Roberts, Riedl and Isbell (2009), the player's overall experience may be more important than the expected payoff. For instance, a hard-fought battle that is lost may be more fulfilling than an easy victory. One aspect of the overall experience of the player is the perception of narrative arc, which can be dynamically generated or adapted with planning techniques.

Indeed, the narrative arc is a crucial aspect of most modern computer games. Game designers use a storyline to lead players through dramatically engaging sequences of events. Role-playing games and other types of contemporary video games usually consist of a series of challenges, or quests, that a player is asked to complete. Rollings and Adams (2003) define gameplay as "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. The story elements provide motivation, set contexts for gaming activities, and propel the game narrative forward. In short, game storylines are used to plan for player experience.

We suggest that optimization of player's experience consists of *presenting the right story to the right person at the right time*. The significance of this claim is twofold. Firstly, game players usually possess diverse motivation, tastes, and needs (Crawford 1984; Yee 2006). A one-size-fits-all script might not be ideal. Secondly, the preferences of players can change over time. After playing one story, they may demand a new one. Therefore, the ability to generate different stories may enhance replayability and improve player experience. Finally, by addressing the first two implications, we are working toward the potential of games that continuously grow and change with the player over a long period of time.

As the cost of labor to write individualized storylines can be prohibitively expensive, AI technologies, specifically planning, can be used to plan for player experience by dynamically generating or adapting storylines in games. In this paper, we concern ourselves with the problem of customizing game narratives for role-playing games while simultaneously maintaining the quality of the aggregated storyline. We acknowledge that computer systems are not capable of the same levels of creativity as humans. Consequently, we aim to automatically *adapt* human-authored game narratives, thereby leveraging human authoring skills and creativity while at the same time scaling up the ability to deliver unique, customized game experience to individual players.

In this paper, we justify our stance of representing game narratives as plans, and discuss the notion of narrative coherence as heuristic features of the plan structure. We then present a technique to adapt and customize human-authored game narratives consisting of game quests. A refinement search algorithm is used to iteratively modify a pre-existing storyline plan until it more closely matches the motivation, tastes, and needs of the target user. Our system is capable of (1) generating a large variety of quest combinations to suit the need of each individual player and enhance re-playability (2) maintaining story quality by maintaining narrative coherence in addition to soundness and (3) balancing the preservation of the original stories and the adaptation to leverage human creativity. The storyline is adapted by adding, deleting or replacing quests. In addition, quests themselves can be altered in content and structure to fit the aggregated storyline.

The remainder of the paper is organized as follows: We discuss the theoretical aspects about narratives, the adaptation problem and the notion of narrative coherence in the next section. After that, we deal with the practical side of narrative adaptation and present the planning algorithm and a detailed example. The fifth section analyzes the theoretical authorial leverage our system empowers game designers with. The last section concludes this paper.

Planning and Narrative

We focus on the narrative aspect of experience, a sequence of events with continuant subject and that constitutes a whole (Prince 1987). In the case of our work, the narrative is a description of the expected sequence of events that will occur in a virtual environment. Following others (c.f., Young 1999; Riedl and Young 2004; Riedl 2009), we computationally represent narratives as a plan. 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. (1991) and Trabasso and van den Broek (1985) in particular highlight the importance of causalities in stories.

Cognitive science and neuroscience suggests that planning may be a very appropriate computational means for narratives. Young and Saver (2001) note that dorsolateral prefrontal injuries simultaneously impair behavioral planning and the ability to produce "narrative account of their experience, wishes and actions" while many other cognitive abilities remain intact. This coincidence seems to hint on the functional similarity of planning and narrative generation in the human brain. Rattermann et al. (2001) suggested adult human performs planning in an analogous manner to partial-order planning. In summary, planning, especially partial-order planning, seems to bear some resemblance to narrative processing and generation mechanisms utilized by human beings.

Given the above evidence, we represent narrative as partially ordered plans. A partial-order plan consists of actions and temporal and causal relations. Actions encode preconditions – conditions that must be true for the action to be executable – and effects – conditions that become true once the action completes. Causal links, denoted

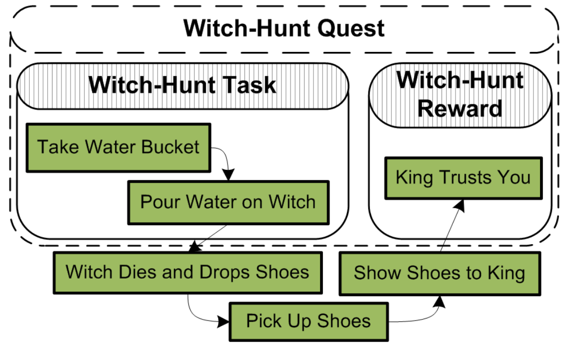

Figure 1 An Example Quest

$a_1 \rightarrow^c a_2$, indicate that the effects of action a_1 establish a condition c in the world necessary for action a_2 to execute. Temporal links indicate ordering constraints between actions.

We use additional representational structure provided by decompositional partial order planning (DPOP) (Young and Pollack 1994). In DPOP, abstract actions are decomposed into more primitive actions using decomposition recipes. Figure 1 is an example of nested decomposition recipes. Rounded rectangles are abstract actions and ordinary rectangles are primitive actions. Encapsulation represents decomposition recipes and arrows represent causal links. For clarity, no temporal links are shown. Primitive actions outside the decompositions are not part of the definition of the decomposition recipe, but are necessary for causal soundness.

The Narrative Adaptation Problem

The conventional planning problem is to find a sound sequence of actions that transforms the world from an initial state into one that satisfies a goal situation. The soundness guarantees correct execution in the absence of uncertainty. The narrative generation problem, in comparison, can be defined as finding a sound and *coherent* sequence of events that narrates the transformation of the world, which involves events, or sequences of events, of significant interest to the audience. Narrative generation contrasts with conventional planning in that the entire experience replaces the final outcome as the primary concern. For example, a tragedy or thriller relies more on relationships between actions in the narrative plan and less on its outcome.

This paper deals with a problem that is slightly different: the *adaptation* of narratives. Instead of generating a narrative from scratch, adaptation starts with a sound and coherent narrative and modifies it to meet the user's requirements. In this paper, we start with a human-authored game narrative. Our algorithm preserves as much as the original narrative as possible to minimize the chance any handcrafted aesthetic or intuitive elements are broken because the algorithm is incapable of understanding them.

Quest-centric game adaptation applied narrative adaptation to computer games in which the main storyline is comprised of one or more challenges. Quests capture the

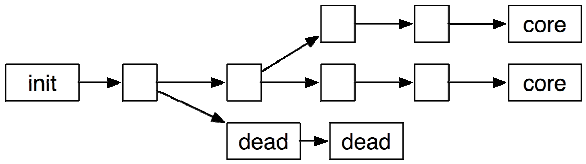

Figure 2 Schematic of dead-end events

different types of challenges in a library of decomposition recipes. We represent quests as decomposition recipes. Figure 1 shows a quest. Quests are decomposed into a task and an award, which allows quests to be reconfigured to fit into a larger storyline.

Narrative Coherence

We believe partial-order plans are effective representations of stories. However, conventional planning algorithms are 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 (1985) 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 preconditions are on causal chains back to the initial state, coherence is achieved when each event has at least one effect on a causal chain to the outcome state. In this section, we distinguish two types of story flaws that break narrative coherence: *dead ends* and *superfluous efforts*. 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.

Core Set. First, we suggest that some events in a story are of special interest to the audience and are 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. These events form the core set of the story. The core set depends on the application. In this paper, we define the core set to include only propositions in the goal state of the plan. However, depending on the circumstances, one may want to choose other events for the core set. For example, to represent complex authorial intent a plan may have intermediate goals (Riedl 2009).

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. For example, when the player is interested in becoming filthily rich, the event where treasures are obtained is crucial, and other events should be subordinate. If the player obtains a magic sword that is not necessary for finding the treasures, then the event of obtaining the sword is a dead end. See Figure 2 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.

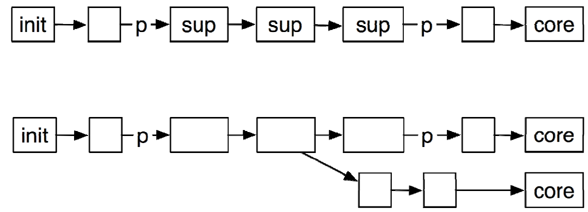

Figure 3 Schematic of superfluous efforts (top) and non-superfluous efforts (bottom).

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 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:

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

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

Superfluous Efforts. Another breach of narrative coherence occurs when events unnecessarily negate and then restore world states. 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. Figure 3 (top) shows superfluous effort because the events serve no purpose other than re-establishing condition p. Figure 3 (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 $S \subseteq V \setminus \{\text{initial, goal}\}$ is a superfluous effort if

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

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.

Planning Algorithm for Adaptation

To optimize a player's experience in a game, we argue that the main storyline of the game should be customized to the player's interests, needs, and motivations. In doing so, we can also produce numerous variations of a finite number of human-crafted storylines, resulting in greatly improved replayability. Figure 4 shows the architecture of our system. The game adaptation process takes a main storyline, a library of quest structures, and a set of player requirements. The player requirements can be provided by the player him or herself or derived from a player model. Currently, we allow players to directly specify their requirements.

Our storyline adaptation algorithm extends DPOP. In the search for the sound plan, DPOP satisfies open preconditions, eliminates causal threats, and decomposes abstract actions until a sound plan is constructed. To fix each flaw, a choice from several strategies is made. By doing so, decompositional partial order planning provides two assurances:

1) All abstract actions are decomposed to primitive actions.
2) Preconditions of actions are satisfied with effects of other actions or the initial condition, and the causal links are not threatened. Indeed, every action is on a causal chain beginning with the initial state.

Nevertheless, standard DPOP is not sufficient to solve the narrative adaptation problem as described above. While retaining the refinement search paradigm, we empower DPOP with the following preprocessing step (1) and extra capabilities (2-3):

1) Modify plan goals. Plan goals reflect authorial intent. As players' requirements change, goals satisfying these requirements must change accordingly.
2) Delete events. While DPOP can only add actions and links to a plan, the adaptation algorithm must be able to delete quests and world-level events that are no longer necessary or that cause the planner to fail.
3) Ensure narrative coherence. Conventional planning techniques only ensure actions' preconditions are on causal chains leading back to the initial state. Narrative coherence requires that actions' effects are on causal leading to important and recognizable outcomes.

The algorithm must delete actions because they represent quests that are not desired as part of the player's experience. Deletion of actions can cause dead ends. Deletion of actions, followed by addition of other actions, can result in superfluous efforts. Consequently, extra effort must be performed to identify and restore narrative coherence.

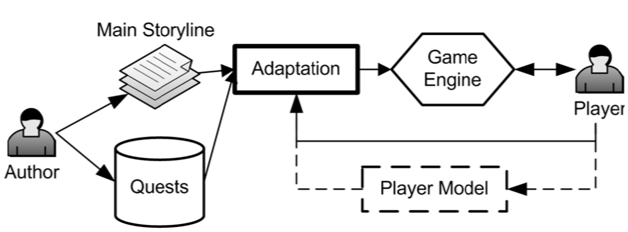

Figure 4 The Storyline Adaptation Pipeline

Customization of the game storyline starts with the modification of the goal state of the original plan according to the set of player requirements. The player can specify a quest selection. When a quest is added to the story, the condition `quest-complete (quest-X)` is added to the goal state as a quest-level goal. As open preconditions are repaired, the quest action will be added to the plan to satisfy the goal. When a quest is no longer desired, its only effect will be removed from the quest-level goal state, causing it to become a dead end and be removed during planning. The goal-state modification is a one-time preprocessing step executed before planning.

Deletion of events is handled as follows. If the event is a primitive event that cannot be further decomposed, it is simply removed along with any causal links and temporal ordering constraints. If the event is part of decomposition, then all other sibling events in the decomposition are also deleted and the parent abstract event is marked as un-decomposed. The rationale is that actions in the same decomposition recipe are considered having some cohesion, and deleting them all allows the abstract action to be re-decomposed. If the event to be deleted is abstract, it is

---

The adaptation algorithm takes a plan structure, a set of flaws evidencing why the plan cannot be a solution, and a domain model consisting of un-instantiated actions.

1. Termination If the plan is inconsistent, fail. Otherwise, if the plan is complete, return.

2. Plan Refinement Choose a flaw from the plan. Switch on flaw type:

- Open Precondition: non-deterministically choose
  - ❖ Reusing an action with a unifying effect
  - ❖ Adding a new action with a unifying effect
  - ❖ Remove the action with the open pre-condition
- Causal Threat: non-deterministically choose promotion, demotion, or deleting the action threatening the link.
- Abstract Action without Decomposition: non-deterministically choose a decomposition from the library and insert actions in the decomposition into the plan, or reuse existing actions as part of the decomposition
- Dead End: non-deterministically choose to do one of the following
  - ❖ Satisfy Precondition: Link one of the dead-end action effects to a unifying open precondition.
  - ❖ Replace Link: Replace a causal link to a unifying precondition with a link from the dead end action.
  - ❖ Remove Action: Remove the action from the plan.
  - ❖ Do Nothing. Ignore the flaw.
- Superfluous Effort:
  - ❖ Link effects of earlier steps to pre-conditions fulfilled by actions in the superfluous effort
  - ❖ Do Nothing. Ignore the flaw.

3. Recursion

Figure 5 Quest-Centric Adaptation Planning

removed along with all children in its decomposition.

The ability to delete and add actions creates circumstances in which the planner revisits the same partial plan in the search space. To preserve *systematicity* of the search and prevent infinite loops, we mark every action and causal link added during the adaptation as "sticky" and do not allow them to be deleted henceforth. Note that actions in the original plan to be adapted are not sticky. The stickiness of actions is a hard constraint that cannot be violated.

The planning algorithm itself is extended in order to process existing plans. The algorithm is shown in Figure 5. There are five types of flaws: (a) unsatisfied precondition of an action, (b) un-decomposed abstract action, (c) causal threats, (d) dead ends and (e) superfluous efforts. Here, we only highlight the differences between our algorithm and a typical partial-order planner. In our algorithm, an action can be deleted due to an open precondition. It is necessary in situations where the precondition cannot be satisfied due to the deletion of other parts of the plan. Similarly, causal threats can be resolved by deleting the action that threatens the causal link. This can be handy in situations where events that have become unnecessary are preventing plan soundness. In both cases, deletion is used with caution.

Two new types of flaws, dead ends and superfluous efforts, are introduced in order to maintain narrative coherence. Normally, dead ends only occur as a result of deletion or during repairing another dead end. We propose four ways to fix a dead end, listed in decreasing order of desirability. The planner can (1) link one effect of the dead end to an open precondition in the plan. Alternatively, it can (2) replace an existing causal link with a link from the dead-end action, the two links satisfying the same pre-condition. In this case, bookkeeping is necessary to prevent infinite loops when two actions are competing for the same link. In addition, the planner can (3) remove the dead end. Finally, if all else fails, we (4) ignore the flaw and accept that the final solution will have a dead end. We believe that this is preferable to failing to return any solution.

Superfluous efforts can be generated by connecting existing actions in an undesirable way, for example, in order to repair a dead end. Also, it may be caused by the removal of some actions the superfluous efforts lead to. To repair a superfluous effort, the algorithm can replace outgoing links from those actions in the superfluous effort with effects from earlier actions. Extending earlier causal effects to later links is a common technique used in continuous planning (Russell and Norvig 2002). As with dead ends, we ignore the flaw if there is no other way to solve the problem, favoring a plan with superfluous actions over no solution.

Example

In this section, we briefly 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 quest, an intermediate step, and the final result are shown respectively in Figure 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.

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

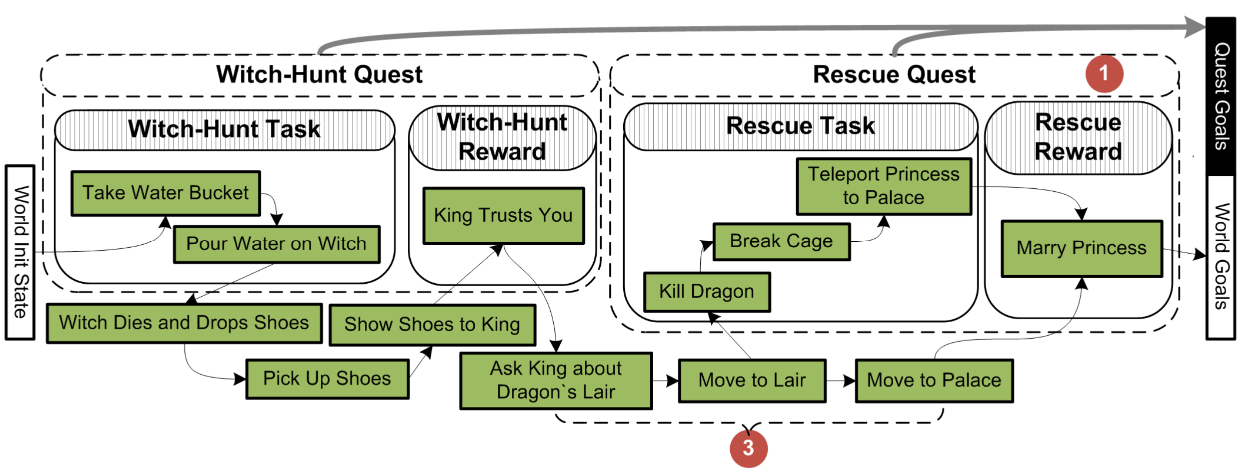

Figure 6 Original Game Narrative Before Adaptation

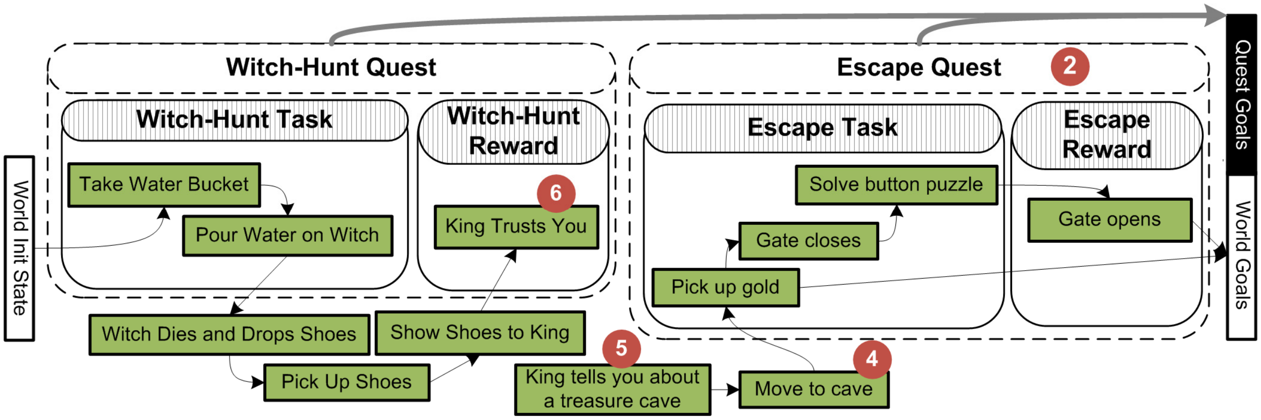

Figure 7 An Intermediate Snapshot in the Adaptation

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 Figure 7.

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 Figure 8.

Authorial Leverage

A direct motivation of this research is to scale up the ability to deliver a large number of 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. In this section, we focus on variability, or the number of distinct stories.

Two components must be authored in the system: a world domain model, containing specifications for primitive event actions as well as a number of quests as DPOP recipes. These constitute a one-time authoring cost by a domain engineer. Next, one or more storylines may be authored in the DPOP representation such that they consist of some quests and other primitive events. This may require additional effort on the part of the human author, but the payoff for this extra effort is an exponential scaling of the initial effort.

Theoretically, our adaptation process takes a single narrative and produces as many adaptations as the size of the power set of available quests. In practice, the number of pragmatic adaptations will be lower because it's likely

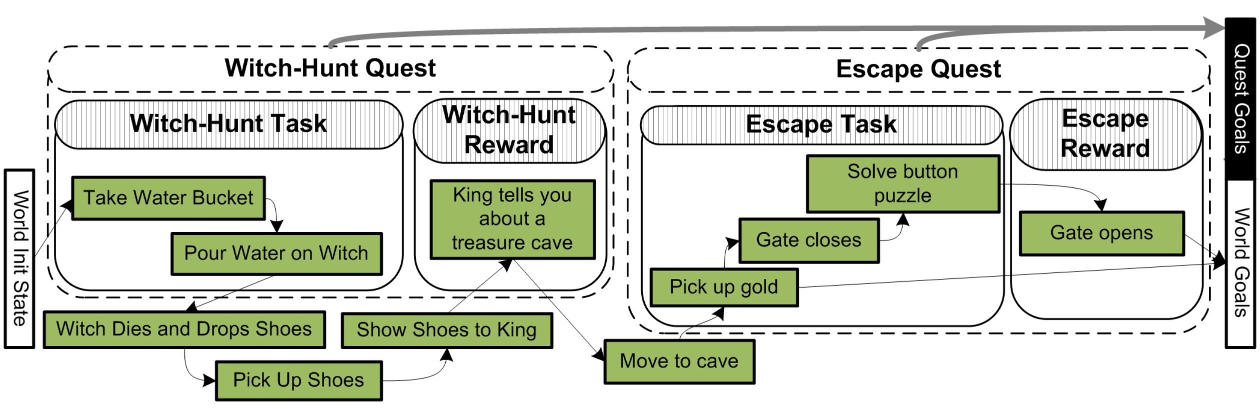

Figure 8 Complete, Coherent Narrative after Adaptation

that a large fraction of the original is retained in each adaptation request. It is possible, for example, that the output story always contains more than 70% of original quest. However, the scaling will still be exponential when 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). However, as the planner can opportunistically discover feasible transitions based on existing actions in the library, the actual authoring effort required can be significantly less. In either case, authoring efforts grows much slower than distinct stories.

Future work is required to measure the pragmatic authorial leverage of the system in terms of authoring effort versus effective output. An evaluation of aesthetic quality of generated storylines is also currently underway.

Related Work

As an offline procedure, storyline 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 planning as the underlying mechanism for selecting and instantiating narrative events (c.f., Meehan 1976, Lebowitz 1987, Riedl and Young 2004, Porteous and Cavazza, 2009). The distinction between our storyline adaptor and story planning is that the storyline adapter starts with a complete, sound narrative structure and is capable of removing events.

Case-based reasoning (CBR) has also been used to generate stories from scratch (c.f. Turner 1994; Peréz y Peréz and Sharples, 2001; Gervás et al. 2005; Turner 1994. Our system may also be compared with the revision stage of transformational case-based planning (CBP), in which old plans are reused and revised to solve new problems. One important difference between CBP and quest-centric adaptation is that planning is geared towards maximum efficiency whereas the shortest or most efficient sequence of actions is rarely the best or most coherent story. Transformational CBP often tries to eliminate as many actions as possible, but we always try to preserve the original narrative and authorial intent if possible. Our search heuristic always favors fixing plan flaws in alternative ways before deleting any actions because deletion of events may interfere with authorial intent.

In a parallel effort, the TACL system (Niehaus and Riedl 2009) 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. We demonstrate direct modification of game quests.

Interactive storytelling systems demonstrate how players or learners may interact with story and scenario content in complex simulation environments. See Roberts and Isbell (2008) for overviews of interactive storytelling and drama management systems. The distinction between quest-centric game adaptation and interactive storytelling is that in interactive storytelling adjustments to the virtual world occur at execution time in order to cope with the real-time actions of the player. Our adaptation system, on the other hand, rewrites the objectives of the virtual environment in an offline process. In this light, quest-centric game adaptation and drama management are complimentary: the adaptation system configures the drama manager, which oversees the user's interactive experience.

While discussion of the execution of game narratives is outside the scope of this paper, we envision using planning-based interactive narrative systems such as the Automated Story Director (Riedl et al. 2008) for dynamic execution of game narratives within the game world.

Work on optimizing player experience in games have been addressed in terms of interactive storytelling. Thue et al. (2007) describe a player modeling approach to choosing different trajectories through *pre-authored* branching story structures based on player profiles. Hullett and Mateas (2009) 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 investigating optimization of player experiences have explored game world generation and other non-narrative content generation using neural network models of players and evolutionary computation (c.f., Togelius et al. 2010). At this moment, we are ignoring the generation of landscape and environment in games.

Conclusions

As game players possess different motivations, tastes and preferences, adapting and customizing game content may improve gaming experiences. In this paper, we presented a planning technique that adapts game storylines consisting of several quests in order to deliver customized narratives for individual players. We proposed two plan structures that break narrative coherence of a story and extend decompositional partial-order planning to repair these coherence flaws and delete actions to adapt existing plans. Furthermore, we discussed design of search heuristics and illustrated the algorithm with a concrete example. The user study suggests the ability to repair dead end is useful in generation of game narratives.

References

Balla, R.-K. and Fern, A. 2009. UCT for Tactical Assault Planning in Real-Time Strategy Games. *Proceedings of the 21st International Joint Conference on Artificial Intelligence.*

Chen, S., Nelson, M.J., Sullivan, A., and Mateas, M. 2009. Evaluating the Authorial Leverage of Drama Management. *Proceedings of the AAAI Spring Symposium on Intelligent Narrative Technologies II*.

Chung, M., M. Buro, and Schaeffer, J. 2005. Monte Carlo Planning in RTS Games. *Proceedings of the 2005 IEEE Symposium on Computational Intelligence and Games*.

Crawford, C. 1984. *The Art of Computer Game Design*. Berkeley, CA: Osborne/McGraw-Hill.

Graesser, A. Lang, K.L., and Roberts, R.M. 1991. Question Answering in the Context of Stories. *Journal of Experimental Psychology: General*, 120(3), 254-277.

Gervás, P., Díaz-agudo, B., Peinado, F. and Hervás, R. 2005. Story plot generation based on CBR, Knowledge-Based Systems, 18(4-5), 235-242.

Hanks, S. and Weld, D. S. 1995. A Domain-Independent Algorithm for Plan Adaptation. *Journal of Artificial Intelligence Research*, 2, 319-360.

Hullett, K. and Mateas, M. 2009. Scenario Generation for Emergency Rescue Training Games. In *Proceedings of the Fourth International Conference on the Foundations of Digital Games*.

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

Meehan, J. 1976. *The Metanovel: Writing Stories by Computer*. Ph.D. Dissertation, Yale University.

Niehaus, J. and Riedl, M. O. 2009. Scenario Adaptation: An Approach to Customizing Computer-Based Training Games and Simulations. In *Proceedings of the AIED 2009 Workshop on Intelligent Educational Games*.

Orkin, J. 2005. Agent architecture considerations for real-time planning in games. *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference*.

Pérez y Pérez, R. and Sharples, M. 2001. MEXICA: A Computer Model of a Cognitive Account of Creative Writing. Journal of Experimental and Theoretical Artificial Intelligence, 13, 119-139.

Porteous, P. and Cavazza, M. 2009. Controlling Narrative Generation with Planning Trajectories: The Role of Constraints. *Proceedings of the 2nd International Conference on Interactive Digital Storytelling*.

Prince, G. 1987. *A Dictionary of Narratology*. University of Nebraska Press.

Rattermann, M.J., Spector, L., Grafman, J., Levin, H., and Harward, H. 2001. Partial and Total-Order Planning: Evidence from normal and prefrontally damaged populations. *Cognitive Science*, 25, 941-975.

Riedl, M.O. 2009. Incorporating Authorial Intent into Generative Narrative Systems. *Proceedings of the AAAI Symposium on Intelligent Narrative Technologies II*.

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 Systems Science and Applications*, 4(2).

Riedl, M.O. and Young, R.M. 2004. An Intent-Driven Planner for Multi-Agent Story Generation. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*.

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), 61-75.

Roberts, D.L., Riedl, M.O., and Isbell, C.L. 2009. Beyond Adversarial: The Case for Game AI as Storytelling. In *Proceedings of the 2009 Conference of the Digital Games Research Association*.

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

Russell, S. and Norvig P. 2002. Artificial Intelligence: A modern approach. Prentice Hall.

Sánchez-Ruiz, A, Lee-Urban, S., Munoz-Avila, H., Diaz-Agudo, B., González-Calero, P. 2007. Game AI for a Turn-based Strategy Game with Plan Adaptation and Ontology-based retrieval. In *Proceedings of ICAPS 2007 Workshop on Planning in Games*.

Smith, S.J.J., Nau, D.S., and Throop, T. 1998. Computer bridge: A big win for AI planning. *AI Magazine* 19(2).

Thue, D., Bulitko, V., Spetch, M., and Wasylishen, E. 2007. Interactive Storytelling: A Player Modelling Approach. *Proceedings of the 3rd Conference on Artificial Intelligence and Interactive Digital Entertainment.*

Togelius, J. Yannakakis, G., Stanley, K., and Browne, C. 2010. Search-based Procedural Content Generation. *Proceedings of the EvoStar Conference.*

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

Turner, S. R. 1994. The Creative Process: A Computer Model of Storytelling. Lawrence Erlbaum Associates.

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, K., and Saver, J.L. 2001. The Neurology of Narrative. *SubStance: A Review of Theory and Literary Criticism*, 30, 72-84.

Young, R.M. 1999. Notes on the Use of Plan Structures in the Creation of Interactive Plot. In *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. *Proceedings of the Second International Conference on Artificial Intelligence and Planning Systems*.