

# Automated Scenario Adaptation in Support of Intelligent Tutoring Systems

**James Niehaus**

Charles River Analytics, Inc.  
Cambridge Massachusetts, USA  
jniehaus@cra.com

**Boyang Li and Mark O. Riedl**

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

## Abstract

Learners may develop expertise by experiencing numerous different but relevant situations. Computer games and virtual simulations can facilitate these training opportunities, however, because of the relative difficulty in authoring new scenarios, the increasing need for new and different scenarios becomes a bottleneck in the learning process. Furthermore, a one-size-fits-all scenario may not address all of the abilities, needs, or goals of a particular learner. To address these issues we present a novel technique, *Automated Scenario Adaptation*, to automatically “rewrite” narrative scenario content to suit individual learners’ needs and abilities and to incorporate recent changes from real world learning needs. Scenario adaptation acts as problem generation for intelligent tutoring systems, producing greater learning opportunities that facilitate engagement and continued learner involvement.

## Introduction

Research shows that expertise is gained through experience and reflection (Katz, Allbritton & Connelly 2003). In the domains of higher-order cognitive skills such as problem-solving, decision-making, and situational awareness, lessons acquired through experience become tacit knowledge – hard to articulate, but highly effective when put into practice (Hedlund, Antonakis, & Sternberg 2002). Expertise is a consequence of exposure to a greater number of varied experiences from which to compile tacit knowledge, and practice under different circumstances is key to generalization of skills and knowledge.

Computer-based training games and simulations are an important part of the equation for generating more adaptive leaders because they can be used in informal learning environments (e.g., the home or in the field), affording more frequent learning opportunities. However, if

repetition under varied circumstances is key, then how can training scenarios be made to present these varied circumstances without causing scenario authoring to be overly tedious or difficult? Complicating the picture, learners may have different needs and abilities, and any given scenario may not be the most effective for each learner. Furthermore, new challenges and situations in the real world may arise, rendering old scenarios obsolete.

We address these challenges in the context of scenario-based training, in which a learner is tasked with completing a mission – a simulated period of time in which situations arise that require the learner to recognize the need to perform a skill and then perform it effectively. Scenarios are correlated with narrative; both “narrative” and “scenario” are descriptions of events.

A typical solution to scenario creation is to deploy authoring tools that enable subject matter experts, trainers, and educators to manually construct new scenarios. Although some examples authoring tools have improved authoring of tutoring systems in non-scenario-based domains (e.g., Aleven et al. 2009), scenario authoring remains difficult, requiring pedagogical knowledge, domain understanding, and storytelling creativity. Manual scenario authoring does not scale well to individually personalized scenarios or to rapid events in the real world that produce new and modified training needs.

The human scenario author is the bottleneck in the pipeline to provide greater numbers of scenario-based learning experiences, more personalized learning experiences, and more manual scenario authoring. It is a bottleneck in the delivery of individualized and highly contextualized scenario-based training and education. By overcoming this bottleneck, training effectiveness can be increased: targeting the individual learner entails incorporating just-in-time information about his or her needs, abilities, motivational styles, and emotions to achieve maximal gain in the available training time.

Though complete automation presents an ideal solution, current AI does not rival human author understanding and creativity. Therefore, this paper proposes a hybrid human-

automation approach, *Automated Scenario Adaptation*. Automated Scenario Adaptation is a process by which an intelligent computational system takes a manually authored scenario and a set of learning objectives customized to an individual, and “rewrites” the scenario, personalizing the content of the scenario. The personalized scenario can then be executed in a virtual game or simulation environment that affords some freedom of action for the learner. Automated Scenario Adaptation is complimentary to intelligent tutoring systems that can identify new training needs and guide the learner through the novel scenario.

## Background and Related Work

Individualized learning for computer-based instruction has been explored in the realm of Intelligent Tutoring Systems (ITS) (Koedinger et al. 1997). ITSs employ a model of the student to generate hypotheses about student proficiency. The general behavior of an ITS can be described as a process with two nested loops (VanLehn, 2006). The outer loop is one of task selection, determining what problem the student attempts next. The inner loop traces the student’s progress through a task and provides appropriate feedback. Automated Scenario Adaptation is complimentary to ITS in the sense that it acts as a problem generator for an individual learner. Scenario adaptation has the potential to increase the effectiveness of tutoring systems. When the scenario content is fixed, an ITS is limited in its ability to achieve beneficial change; the scenario content may not be sensitive to the needs. Parts of the overall scenario may be too easy or too difficult, unnecessary due of learner proficiency in certain aspects, or require skills that the learner has not yet acquired.

Because of the correlation between “narrative” and “scenario”, adaptation of scenario content in virtual environments has been explored in the context of *interactive narratives*. An interactive narrative system dynamically changes future events in the virtual world in response to real-time user actions, and techniques for modifying narrative content to meet authorial goals may be employed to modify scenario content to meet training goals. Riedl et al. (2005) explore the relationship between interactive narrative and intelligent tutoring.

There are several examples of interactive narrative systems for training. The INTERACTIVE STORY ARCHITECTURE FOR TRAINING (ISAT) (Magerko et al. 2006) reactively selects the next episode in a training simulation based on current performance by the learner. Logical forms ensure causal plausibility from episode to episode. Episodes are continuously selected and executed until a complete set of skills has been practiced. ISAT further manipulates the behaviors of NPCs to give hints or to draw attention to salient aspects of the task at hand. Likewise, the interactive narrative aspects of CRYSTAL ISLAND (Mott and Lester 2008) employ probabilistic reasoning over future events to reinforce the learner’s use of the scientific method. The AUTOMATED STORY DIRECTOR (Riedl et al. 2008), conversely, takes an existing

scenario and then generates branches whenever the learner performs an action that threatens the causal coherence of the remainder of the scenario. The system ensures that certain relevant learning situations will unfold regardless of how the learner chooses to go about resolving dilemmas. The technique described in this paper may be used to generate the initial story structure used by the systems discussed above. AEINS (Hodhod et al. 2010) opts for a reactive approach, using an ITS to select the next “teaching moment” and then employing a planner to generate transitions from the current state to the teaching moment.

Real-time story adaptation – e.g., interactive narrative – necessarily limits computation time and is prone to local maxima based on the history of events in the training session. Offline scenario generation may more readily manipulate the global structure of learning content, potentially achieving more optimal scenario content. Hullett and Mateas (2009) present an offline approach to constructing novel practice environments for emergency management. In the context of entertainment, story generation is the problem of producing a narrative from scratch. Story generation systems commonly use planning to select the narrative events (Lebowitz 1985; Porteous and Cavazza 2009; Riedl and Young 2010). The system described in this paper is an offline adaptation system that attempts to optimize a scenario based on a learner model. It is thus more closely aligned to story generation systems although, as noted above, is complimentary with many interactive narrative systems.

## Scenario Adaptation

Scenario adaptation is a process designed to augment the ability of a single human scenario author to deliver personalized learning experiences to numerous individual learners. The key advancement is to inject automation into the traditional scenario-authoring pipeline. The traditional scenario-authoring pipeline is shown in Figure 1:

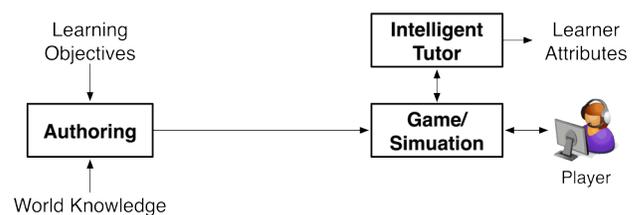


Figure 1. The traditional scenario-authoring pipeline.

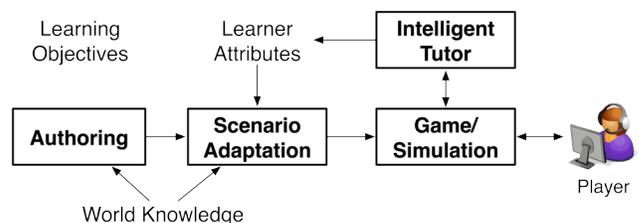


Figure 2. The scenario adaptation architecture.

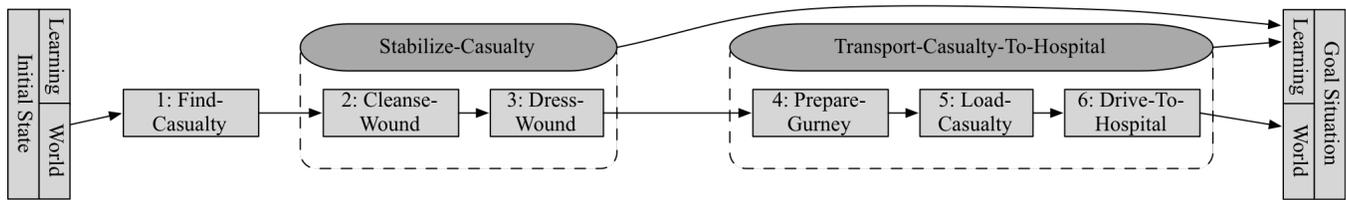


Figure 3. A simple manually authored scenario plan.

scenario is manually authored and then played by a learner in a virtual game or simulation environment. A tutor may be present to track the learner's progress in the simulation and record changes to the learner's attributes. In contrast, our scenario adaptation architecture is shown in Figure 2. The cycle from simulation to intelligent tutor to scenario adaptation enables a single manually authored scenario to be recycled by progressively adapting to the learner's changing needs and abilities.

In this paper, we focus on the scenario adaptation process. By "rewriting" manually authored scenarios, our system leverages human intuition to create a theoretically exponential number of unique experiences. Leverage, in this case, means capitalizing on human effort to provide an increased number of experiential opportunities (Chen et al. 2009).

Modifications to the human authored scenario may take the following forms:

- Adding events and elements to provide additional opportunities to practice a certain skill, or to address an additional learning objective.
- Removing events and elements when the learner is already proficient.
- Changing the details of events to make them easier or harder.

For automated scenario adaptation, we must address two questions: (1) how to represent scenarios, and (2) how does a computational system reason about and manipulate scenario content using knowledge about the learner?

## Scenario Representation

Following previous approaches (Young 1999; Riedl et al. 2008; Porteous and Cavazza 2009; Riedl and Young 2010), we employ plan-like representations of narrative. Plan-like representations capture causality and temporality of action and provide a formal framework built on first principles, such as soundness and coherence, for selecting and ordering events. However, unlike a plan meant for execution by a single entity, we use plans as descriptions of events expected to unfold in a virtual world. The original scenario must be made out of the following structures:

- **Events.** An event is a specification of an occurrence that affects significant change on the world. Events have preconditions – conditions in the world that must be true for the event to occur – and effects – conditions in the world that are different after the event completes.
- **Learning objectives.** A learning objective is an abstract

event that, in addition to the normal event structure, aggregates several primitive events that relate to a measurable skill.

- **Temporal constraints.** Temporal constraints describe a partial ordering over events. Events unordered with respect to each other may happen concurrently in the world.
- **Causal links.** A causal link describes causal relations between events where the effects of one event establish condition in the world necessary for a successive event to occur.

A simple example scenario plan for a cultural negotiation mission is shown in Figure 3, for decision and problem solving training of Non-Governmental Organization emergency medical personnel in civil war regions. Boxes represent events. Note that the actual execution of any event may involve several successive actions on the part of the learner and non-player characters. Ovals represent learning objectives with the encapsulation relationship indicating which events correspond to which learning objectives. Not all events need be related to a learning event; *connective events* provide narrative coherence by linking events in one learning objective to those in another.

The original scenario plan is an input to the scenario adaptor. Additionally, the scenario adaptor requires domain knowledge in the form of a *domain theory*. The domain theory describes the way in which a particular world can evolve. In our scenario adaptor, the domain theory consists of event templates for all events that could ever possibly happen in the world, and learning objective decomposition rules. Event templates make reference to variables that can be bound to specific instances of entities, such as characters, objects, or places. Learning objective decomposition rules are grammar-like structures that describe legal combinations of events for a learning objective and can either be very specific, enumerating an exact set of events that must exist in the scenario when the learning objective is present, or partial, describing the types of changes to the world state that should occur. Importantly, these decomposition rules provide a guarantee that the scenario adaptor cannot produce scenario structures that are not pedagogically correct. See Young and Pollack (1994) for the form of the rules.

Additionally, the domain theory must provide an initial state and goal situation. The initial state is everything that is true about the virtual world before the scenario begins, including declarations of characters and their roles, places, and objects. The closed world assumption holds for the initial state. The goal situation describes what the scenario,

---

The algorithm takes a scenario plan, an initial state and goal situation, and a domain theory  $A$ .

```
function ScADAPT (plan, init, goal,  $A$ ) returns solution or fail
  fringe  $\leftarrow$  {plan}
  loop do
    if fringe =  $\emptyset$  then return failure
    plan  $\leftarrow$  POP(fringe)
    if plan has no flaws then return plan
    flaw  $\leftarrow$  GET-ONE-FLAW(plan)
    newplans  $\leftarrow$  REPAIR(flaw, plan,  $A$ )
    fringe  $\leftarrow$  INSERT-AND-SORT(newplans, fringe)
```

---

**Figure 4. The plotline adaptation algorithm**

in terms of changes to the world state and anticipated changes to the learner, achieves.

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

### Adaptation Algorithm

The scenario adaptation process is a specialized plan refinement search 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 (*cf.*, Penberthy and Weld 1992) is an example 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. Repairing a flaw may introduce new flaws.

Our plan refinement search algorithm receives as input the following components:

- A complete scenario plan – a partially ordered, hierarchical plan – composed of events within and outside of learning objectives.
- An initial state and goal situation. Of particular importance are the learner state goal conditions indicating skills practiced or situations familiarized.
- A domain theory.

The first thing to note is that there are initially no causal connections from the initial state to events (or learning

objectives), nor causal links from events (or learning objectives) to the goal situation. Thus, although a complete scenario plan is provided as input, the scenario is flawed.

Our adaptation algorithm is shown in Figure 4. We implement the following flaw types:

- **Open condition:** an event has a precondition not satisfied by any causal links from events ordered before or the initial state.
- **Causal threat:** An event has an effect that undoes a condition necessary for another event to occur with no ordering constraints preventing the interaction.
- **Un-decomposed event:** An abstract event has not been decomposed.
- **Dead end:** An event is not on a causal path to the goal situation.

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

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

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

**Subtractive Strategies.** 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

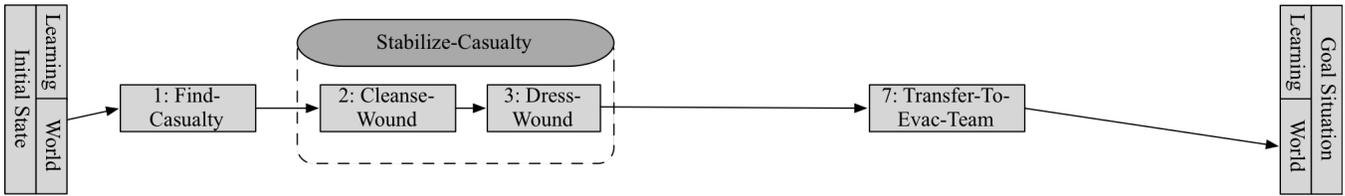


Figure 5. The scenario plan after removing a learning objective.

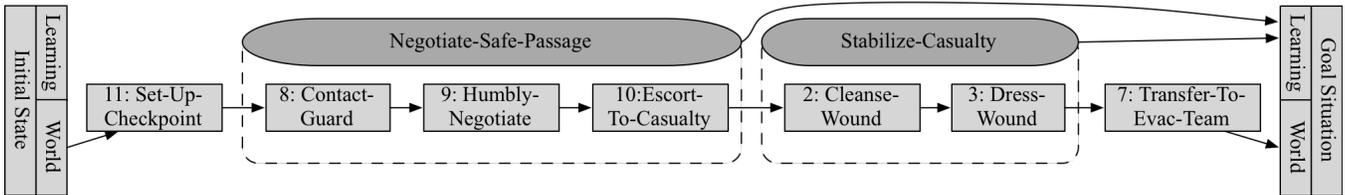


Figure 6. The final scenario plan after adaptation.

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. Deletion may cause new flaws that cause later repair by additive strategies.

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

## Example

The short scenario plan in Figure 3 is has two learning objectives for Non-Governmental Organization emergency medical personnel operating in an area of civil war. The scenario adaptor personalizes it to a particular learner. Assume this learner is proficient at transporting casualties, but needs additional practice negotiating safe passage through militia checkpoints. The goal situation thus enumerates `practiced(stabilize-casualty)` and `practiced(negotiate-passage)` as desired outcomes, as well as a world goal condition that the casualty is returned to the hospital.

First, the `Stabilize-Casualty` learning objective is linked to the goal situation. However, the `Transport-Casualty` learning objective is a dead-end that cannot be linked to the goal situation and is deleted along with all its associated events. The event `Dress-Wound` (3) becomes a dead-end and the condition in the goal situation that the casualty is at the hospital becomes an unresolved. To resolve the goal situation, a new connective event,

`Transfer-To-Evac-Team` (7) is instantiated, which is eventually connected to `Dress-Wound`, resolving `Dress-Wound`’s dead-end status. Figure 5 shows the scenario plan at this point.

The goal condition `practiced(negotiate-passage)` is satisfied by instantiating a new learning objective, `Negotiate-Safe-Passage`. At this time, a decomposition rule is selected, resulting in the instantiation of several primitive events (8-10) in which the learner contacts the checkpoint guard, negotiates passage, and allow him or herself to be escorted to the casualty. However, `Escort-To-Casualty` (10) is a dead-end. This is resolved by moving the link from `Find-Casualty` (1) to `Escort-To-Casualty`, making `Find-Casualty` a dead-end instead. This is finally resolved by removing `Find-Casualty`.

There is one last flaw to consider: where did the checkpoint come from? It is not part of the initial state (it would have been impossible to predict its need). The `Contact-Guard` event (11) requires a checkpoint, which is resolved by instantiating an event in which a checkpoint gets set up before the learner begins the scenario. Once this is resolved, there are no more flaws; the final scenario plan is shown in Figure 6.

## Conclusions

Our scenario adaptation process is capable of taking existing, human-authored scenarios, provided in a particular representation, and adapting them to the needs and abilities of individual learners and to update them to be relevant to changes in the real world. We achieve this by providing customization information into a plan refinement search process modified to be able to add and delete hierarchical scenario content.

Automated Scenario Authoring effectively leverages human scenario authoring ability to provide potentially exponentially many, novel personalized experiences. Scenario adaptation can theoretically produce as many variations of a given plotline as the size of the power set of

available learning objectives. In practice, the number may be lower because a large fraction of the original content will be retained in each adaptation. One of the strengths of scenario adaptation is the ability to opportunistically discover new transitions between learning objectives in order to preserve coherence.

Future work is required to measure the pragmatic authorial leverage of the system, scaling of scenario effort in target domains, and pedagogical effectiveness of adapted scenarios. In other work (Li and Riedl, 2010), we show that a variation of our system designed for personalizing computer role-playing games has favorable qualities with respect to narrative coherence. To provide further evidence of the authorial leverage, experiments are needed that include authoring of training scenarios and measurements of variation. To provide further evidence of pedagogical utility, experiments are also needed that include learners interacting with the training scenarios.

Scenario adaptation is one step toward overcoming the bottleneck of manual content creation for scenario-based computer training games and simulations. By overcoming this bottleneck, learners can be presented with a greater number of unique learning experiences. Those learning experiences can be made highly relevant to individual learners by adapting them to that learners' specific needs and abilities, which in theory can lead to more effective acquisition of expertise in learning domains involving cognitive skills.

## Acknowledgements

The project or effort described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

## References

- Aleven, V., McLaren, B.M., Sewall, J., & Koedinger, K.R. 2009. A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19(2), 105-154.
- Chen, S., Nelson, M. and Mateas, M. 2009. Evaluating the Authorial Leverage of Drama Management. *Proc. of the 5th AI and Interactive Digital Entertainment Conference*.
- Hodhod, R., Cairns, P. and Kudenko, D. 2011. Innovative Integrated Architecture for Educational Games: Challenges and Merits. In *Transactions on Edutainment V*. Springer.
- Hullett, K. and Mateas, M. 2009. Scenario Generation for Emergency Rescue Training Games. *Proc. of the 4th International Conference on the Foundations of Digital Games*.
- Katz, S., Allbritton, D., and Connelly, J. 2003. Going Beyond the Problem Given: How Human Tutors Use Post-Solution Discussions to Support Transfer. *International Journal of Artificial Intelligence in Education*, 13, 79–116.
- Koedinger, K., Anderson, J. R., Hadley, W. and Mark, M. 1997. Intelligent Tutoring Goes to School in the Big City. *International Journal of Artificial Intelligence in Education*, 8, 30-43.
- Lebowitz, M. 1987. Planning stories. *Proc. of the 9th Annual Conference of the Cognitive Science Society*.
- Li, B. and Riedl, M.O. 2010. An Offline Planning Approach to Game Plotline Adaptation. *Proc. of the 6th AI and Interactive Digital Entertainment Conference*.
- Magerko, B. Wray, R., Holt, L., and Stensrud, B. 2005. Customizing Interactive Training through Individualized Content and Increased Engagement. *Proc. of the 2005 Interservice/Industry Training, Simulation, and Education Conference*.
- Mott, B. and Lester, J. 2006. U-Director: A Decision-Theoretic Narrative Planning Architecture for Storytelling Environments. *Proc. of the 5<sup>th</sup> International Conference on Autonomous Agents and Multiagent Systems*.
- Penberthy, J. and Weld, D. 1992. UCPOP: A Sound, Complete, Partial-Order Planner for ADL. *Proc. of the 3rd International Conference on Knowledge Representation and Reasoning*.
- Porteous, P. and Cavazza, M. 2009. Controlling Narrative Generation with Planning Trajectories: The Role of Constraints. *Proc. of the 2nd International Conference on Interactive Digital Storytelling*.
- Riedl, M.O., Lane, H.C., Hill, R., and Swartout, W. 2005. Automated Story Direction and Intelligent Tutoring: Towards a Unifying Architecture. *Proc. of the 2005 AIED Workshop on Narrative Learning Environments*.
- 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. 2010. Narrative Planning: Balancing Plot and Character. *Journal of AI Research*, 39.
- Hedlund, J., Antonakis, J., and Sternberg, R. 2003. Tacit Knowledge and Practical Intelligence: Understanding the Lessons of Experience (ARI Research Note 2003-04).
- Trabasso, T. and van den Broek, P. 1985. Causal Thinking and the Representation of Narrative Events. *Journal of Memory and Language* 24: 612-630.
- VanLehn, K. 2006. The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 16.
- Young, R.M. 1999. Notes on the Use of Plan Structures in the Creation of Interactive Plot. *Working Notes of the AAAI Fall Symposium on Narrative Intelligence*.
- Young, R. M. and Pollack, M.E 1994. Decomposition and causality in partial-order planning. *Proc. of the 2nd International Conference on AI and Planning Systems*.