Scenario Adaptation: An Approach to Customizing Computer-Based Training Games and Simulations

James Niehaus and Mark Riedl
School of Interactive Computing, Georgia Institute of Technology
jniehaus@cc.gatech.edu, riedl@cc.gatech.edu

Abstract. Game-based learning environments often make use of pre-scripted scenarios to present educational and training content. However, a one-size-fits-all scenario may not address all of the abilities, needs, or goals of a particular learner. This paper presents a methodology for automatically adapting a game scenario to better suit such requirements. We describe initial steps toward an intelligent technology called a Scenario Adaptor that employs planning-like algorithms to add or remove events from a scenario that relate to learning objectives. The Scenario Adaptor attempts to "re-write" scenarios in order for the learner to achieve the desired set of learner-specific learning objectives. Scenario cohesion is maintained by integrating causal chains of events that lead to the scenario's outcome. We also briefly discuss scenario execution in two different game environments.

1. Introduction

Training scenarios - especially those utilized by military, intelligence analysts, and emergency responders - are often utilized to give learners hands-on experience with real-life problem solving tasks in game environments. Learners experience a simulated course of events, assess the situation, practice skills, and act to achieve the goals of the scenario. Because the scenario is simulated within the game world, errors can be addressed immediately by an instructor or automated system and there are few repercussions for failure. Thus training scenarios are ideal for task domains where actual failure can be dangerous or costly. Game-based learning environments often utilize scenarios to provide effective training experiences. In these systems, scenarios may be simply a setting for a simulation, or they may be a predefined set of events, a script, that is to occur during the simulation. This work focuses on script-based scenarios.

Training scenarios are often authored by domain and system experts. Not only must the scenario present an accurate picture of the task domain, it must also give the learner opportunities to practice the important learning objectives. Scenarios can be difficult and time consuming to author and implement, and the result of this process is often a small set of one-size-fits-all scenarios crafted for the typical learner. However, such scenarios may not adequately address a particular learner's abilities, needs, or goals. It may force the learner to practice concepts with which he is already quite familiar and ignore concepts which require more practice.

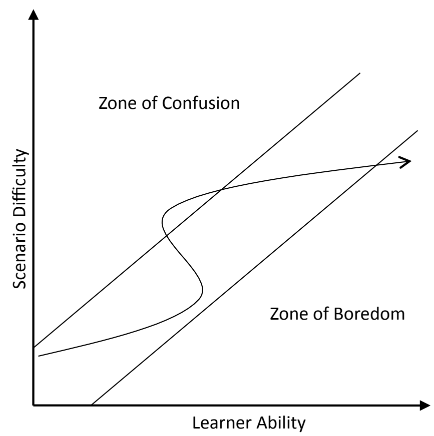

Figure 1. Graph demonstrating the Zone of Proximal Development.

The Zone of Proximal Development (ZPD) (see Figure 1) provides a framework for thinking about a learning experience in training scenarios. The ZPD is the trade-off between learner ability and scenario difficulty. The ideal situation is that, over time, the learner does not stray out of the ZPD into the zone of confusion or zone of boredom. While the ZPD is a developmental theory, it can be applied to training scenarios as well. Here, the ZPD is the difference between the ability the learner has *already demonstrated* and the most difficult challenge the learner is *able to overcome*. A training scenario (e.g., the curved line in Figure 1) is considered effective if falls within the ZPD.

How do we intelligently change scenario data to maximize time spent in the ZPD? We observe that script-based scenarios share many features with stories. Both "narrative" and "scenario" are descriptions of events or sequences of events. Whereas narratives often describe past events, scenarios describe - at some level of abstraction or specificity - events that are expected to unfold. Hence, intelligent narrative technologies such as narrative generation systems [1,2,3,4,5] and interactive narratives [6,7,8,9] may be an embarking point for scenario adaptation.

Interactive narrative systems have been explored as tools for education and training [7,9,4,8]. However, scenario adaptation is not the same as interactive narrative. Interactive narrative systems change the execution dynamics to react to the actions of a user/trainee within the scope of the original system parameters. Scenario adaptation is a process that occurs *before* execution that alters the parameters and the scope of the scenario. Interactive narrative can dynamically change the execution of a scenario, but only Scenario Adaptation can change what the scenario is *about*. The time just prior to scenario execution is the time in which the needs and abilities of the trainee can be most effectively taken into consideration. We view a Scenario Adaptor and interactive narrative systems as complimentary. The scenario adaptor provides the initial, individualized game scenario and the interactive narrative system manages execution in the game environment with the goal of achieving the adapted scenario. This is especially true of the interactive narrative systems based on planning (cf., [6,7]) because they employ similar narrative/scenario representations.

The work described here takes initial steps toward an intelligent system called a *Scenario Adaptor* that automatically customizes a scenario to suit a learner's abilities, needs, or goals. Given a profile of a particular learner, the Scenario Adaptor customizes a scenario to target the ZPD. This allows the learner to practice underdeveloped skills and avoid the redundancy in areas in which she has shown proficiency. We define operations for deleting, adding, and replacing learning objectives in a scenario, we present an extension to partial order planning for scenario adaptation, and we discuss how this process may be used for game-based learning environments.

2. Related Work

Due to our perspective on scenario adaptation as resembling a process of "re-writing" existing scenario content, we note that there is a class of technologies – narrative generators – that may inform scenario adaptation. Though there are many systems that use many techniques for narrative generation [2,3,5,1,8], the systems that are most relevant to our problem of scenario adaptation are those based on planning formalisms. (e.g., [3,1]). Planning based narrative generators are given a goal state specification that must be achieved, and any narrative that does not achieve the goal state is rejected as incomplete. In scenario adaptation, it is necessary to drive a scenario to a goal state because the goal state specifies when the learning objectives are met. Failing to reach the goal state is a failure to meet the learning objectives.

Interactive narrative systems demonstrate how players or learners may interact with story and scenario content in complex simulation environments. While interactive narrative systems may adjust the simulation during execution to meet story or learning goals, scenario adaptation rewrites the goals of the scenario. Thus, we view these technologies as complimentary. The Crystal Island interactive narrative [8] attempts to guide the player through a story by modeling the player and automatically including hints and player aids. The ISAT system [9] is a training system for tactical field medical care that chooses interaction scenes to maximize players practice in related skills. The FearNot! system [4] uses virtual agents to play as characters in a simulation to teach children about school bullying.

The Mimesis system [6] employed a partial-order planner to generate plots for interactive narrative. The Automated Story Director [7] system represented stories as plans and uses a director and character agents to execute the story plans. The director agent to sends commands regarding the story to the character agents that executed these commands. The character agents also possess "local" freedom to act in a natural manner and engage the player. These systems demonstrate the use of plans as stories and the possibilities of mixing a planning approach with character agents.

There is currently much demand for computer-based training for non-kinetic skills such as social awareness, cultural awareness, and bilateral negotiation. These so-called "soft skills" are crucial, and also challenging to train because they occur in ill-defined contexts and situations. Many computer-based training systems targeting soft skills have been developed [10,11,12,13,9,8], and these systems employ scenarios to manage learning content. There is a need for a greater range of scenario contexts and thus non-kinetic skills scenarios are a natural target for scenario adaptation. Each system represents scenarios in a different manner, and the Scenario Adaptor must be sensitive to these representations in order to generate effective scenarios. Section 4 examines the means by which the Scenario Adaptor may be integrated with the game-based environments below.

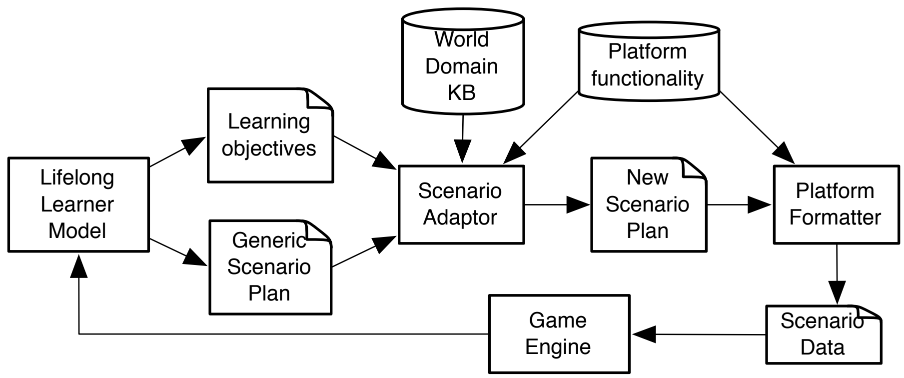

Figure 2. Process for adapting a scenario.

The Adaptive Thinking & Leadership (ATL) program [10] and DARWARS Ambush NK! [11] in particular are a virtual environments to train soft-skills. ATL and NK! feature multiplayer interaction with roles for an instructor and multiple student role players. Scenarios in ATL and NK! are script-based; they are carefully written ahead of time and then managed by the instructor and student role players. The manual process of customizing the scenario may be done for the course material, but the scenario is not reauthored to fit the needs of a particular group of students.

3. Scenario Adaptation

Due to our perspective on scenario adaptation as resembling a process of "re-writing" existing scenario content, our approach to the Scenario Adaptor extends on previous work on narrative generation and interactive narrative systems. Figure 2 depicts the process for adapting scenarios. Two external processes are required. First is a *Lifelong Learner Model* [14], an intelligent system that tracks a learner across many learning experiences (e.g., training scenarios), updating a learner model, and choosing the next training scenario that will help the learner advance. The second is a *Game Engine* that executes a training scenario with the learner as an interactive participant. A generic scenario plan and a set of learning objectives are combined to create a scenario adaptation problem using a world domain knowledge base. The generic scenario plan defines the scenario to be adapted, and the learning objectives define the goals for adaptation. The world domain knowledge base provides a library of possible world events, learning objectives, characters, objects, and other setting related information defining the dynamics of the simulation world and ensuring the adapted scenario is possible within the simulation. The scenario adaptation problem is solved by the Scenario Adaptor, described below, and the resulting scenario is formatted for execution within the game environment. Once the game is played, proficiency is assessed and reported back to the Lifelong Learner Model. See [15] for more about updating the lifelong learner model.

3.1. Scenario and Learning Objective Representation

Two basic requirements influence our representation of scenarios: coherence and pedagogical efficacy. In order for a scenario to be coherent, it must operate within the rules of the simulation. In order for a scenario to achieve pedagogical efficacy, it must provide the opportunity to learn the desired learning objectives. A simulation planning domain is defined to encompass the possibilities for events within the game world and a mapping between event sequences and learning goals is defined to reason about learning objectives.

We employ a specialized plan representation from decompositional partial order planning (DPOP) to represent the scenarios and associated learning objectives [16] (see Figure 4 for a schematic view). DPOP plans contain steps with preconditions and effects. The steps are related via causal links, denoting that one effect satisfies one precondition, ordering constraints, denoting that a step comes before another, and decomposition links, denoting that a step is part of another's decomposition. Abstract steps must be decomposed via decomposition recipes (or, simply, decompositions) to into other abstract steps, non-abstract steps, causal links, and orderings between steps.

To represent both scenario events and learning objectives, we split the planning namespace into world state and learning state. Non-abstract world steps define the events in the scenario, and abstract learning steps define preconditions and effects on the learning state. The scenario initial and goal states contain both world and learner state. The decompositions of the abstract learning level steps are defined by the domain author such that experiencing the steps in the decomposition gives the learner the opportunity to practice the learning objective. Thus, the decompositions define the possible collections of events for achieving learning objectives.

Figure 4 is an example of a simple scenario with two learning objectives. In this example, the learning objective of 'Treat Victim for Arm Injury' is composed of the world steps 'Cleanse Wound' and 'Dress Wound'. The step 'Find Victim' is not part of a learning objective, but it serves the purpose of achieving coherence in the scenario. The victim must be found before he can be treated.

3.2. Adaptation Operations

Given our representation of scenarios and learning objectives, our definition of scenario adaptation is the task of adding, removing, or replacing abstract learning level steps to satisfy learner state preconditions and completing the plan by decomposing all abstract steps and satisfying all preconditions. The starting scenario is assumed to be complete in that all preconditions are satisfied with causal links and no links are threatened. The additional requirement of scenario cohesion indicates that most steps should contribute to a causal chain leading to the goal state (the main scenario outcome). Because learner time is valuable, the scenario should be no longer than required to achieve the learning objectives and promote causal cohesion. We identify three basic operations to adapt a scenario: deletion, addition, and replacement.

3.2.1. Deletion

Deletion is the process of removing a learning objective an its associated steps. Once a learning objective is no longer required by the learning level goal state or other learning level steps, it is deleted. This operation may break causal links, preventing the preconditions of steps from being satisfied and breaking causal chains leading to the goal state. New links, possibly from new steps, must be created to establish preconditions and form chains to the goal state. To delete a learning objective, the learning step and all steps which belong to its decomposition are first removed along with all associated causal links and orderings. This removal may leave an incomplete plan containing link threats, dead end steps, and open preconditions. Planning is performed to complete the plan.

3.2.2. Addition

Addition is the process of adding a learning step and associated world steps. New abstract learning steps may be required to satisfy new preconditions of the learning level goal state or of other learning steps. Addition requires that new world state preconditions are satisfied and new world steps are linked to causal chains to the goal state. To add a learning objective, the learning step and one of its decompositions are inserted into the plan. This insertion may create an incomplete plan containing link threats, dead end steps, and open preconditions. Planning is performed to complete the plan.

3.2.3. Replacement

Replacement is the process of replacing an existing learning level step and its steps with a new learning level step and associated world steps. Replacement is combines deletion and addition. First the existing learning level step and associated world steps are removed, along with all relevant causal links and ordering. Then, the new learning level step and its associated world steps are inserted. Planning is performed to complete the plan.

3.3. Scenario Adaptation Planning

Standard decompositional planning is not sufficient to support the operations in scenario adaptation as defined above. Decompositional planning provides two assurances: 1) that the events relating to the learning objectives are contained in the scenario and 2) that the preconditions of the events are satisfied. However, decompositional planning does not attempt to a) ensure a cohesive causal structure nor b) define methods for adapting existing complete plans. Events that are added as part of a decomposition may not contribute to the goal of the plan, leaving possible causal 'dead ends'. Such dead end steps break the cohesion of the scenario. Decompositional planning is defined in such a way as to only add steps, links, and decompositions to a plan until completeness is achieved. However in scenario adaptation as defined above, a new complete plan must be produced from an existing complete plan, and the removal of steps, links, or decompositions may be required to integrate new events into causal chains leading to the goal.

To satisfy preconditions, maintain plan consistency, and eliminate dead end steps we propose a Scenario adaptation planning algorithm. Figure 3 outlines the algorithm. Scenario adaptation planning performs the standard decompositional planning techniques of satisfying open preconditions, eliminating causal threats, and decomposing abstract steps. The algorithm also attempts to improve plan cohesion by eliminating dead end steps that do not contribute to a causal chain leading to the goal. This algorithm is utilized to complete scenario plans after performing the learning objective operations.

3.4. Example

Figure 4 is an example input scenario featuring a domain for medical field training serious game. The game world for this scenario contains a single victim, lying hurt in some

Scenario Adaptation Planning (problem=⟨plan, learning-objectives⟩, domain-kb)

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:** Resolved in the standard manner via reusing a step or adding a new step to establish the precondition. (cf. [16])
   - **Causal Threat:** Resolved in the standard manner via promotion, demotion, or separation. (cf. [16])
   - **Abstract Step without Decomposition:** Resolved in the standard manner via inserting a decomposition from the library into the plan. (cf. [16])
   - **Dead End Step:** Choose to do one of the following
     - ∗ **Nothing:** Do nothing, leave the step as a dead end.
     - ∗ **Satisfy Precondition:** Link one of the step effects to a unifying open precondition.
     - ∗ **Replace Link:** Replace a causal link to a unifying precondition with a link from the dead end step.
     - ∗ **Remove Step:** Remove the step from the plan, if the step is part of a decomposition, remove the entire decomposition.
3. **Recurse**

Figure 3. Scenario Adaptation planning attempts to incorporate Dead End steps in the causal chains leading to the goal state.

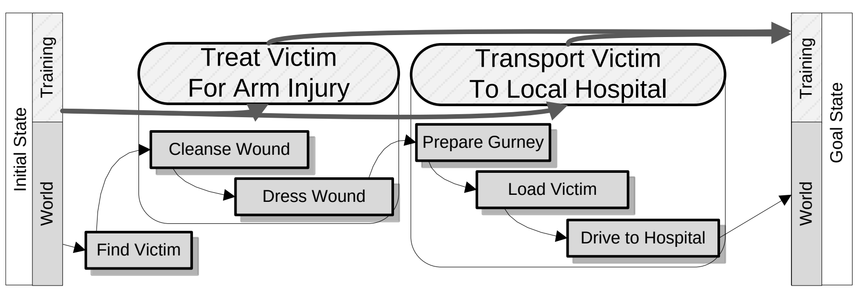

Figure 4. A representation of a simple scenario with learning objectives.

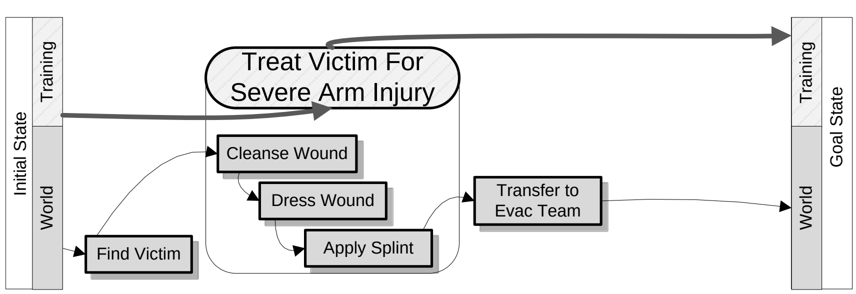

Figure 5. Scenario adapted by removing a learning objective and replacing a learning objective with a more difficult one.

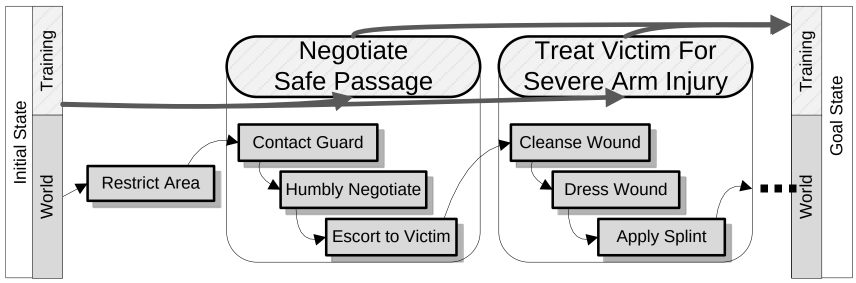

Figure 6. Scenario adapted by adding a learning objective.

wreckage. The learner must locate the victim, treat his injuries, and see him to safety. In this scenario, the learner is given the opportunity to practice two skills, treating a victim for an arm injury and transporting a victim to a hospital. The world steps such as 'Cleanse Wound' denote the actual events in which the learner acts or observes. The learning steps such as 'Treat Victim for Arm Injury' decompose into the world steps that provide the requisite practice. This scenario is an example of be the input, general scenario, given to the Scenario Adaptor.

Perhaps the learner is proficient in transporting victims but wishes to practice more difficult first aid procedures. The first adaptation requests a deletion of the 'Transport Victim to Local Hospital' learning objective and a replacement of 'Treat Victim for Arm Injury' with the more difficult 'Treat Victim for Severe Arm Injury'. As per the deletion operation, the 'Transport Victim to Local Hospital' step and associated world steps and links are removed. As per the replacement operation, the 'Treat Victim for Arm Injury' step and associated world steps and links are removed, and a new 'Treat Victim for Severe Arm Injury' step is added. The planner is then invoked to repair the scenario. The results are depicted in Figure 5. Here, the learner locates the victim, treats a severe arm injury, and then a computer controlled evacuation team arrives to transport the victim to safety.

The learner may require practice in negotiation. The second adaptation requests an addition of the 'Negotiate Safe Passage' learning objective. The results of this adaptation are in Figure 6. Before the learner finds the victim, a computer controlled officer restricts the wreckage area. The learner must arrive on the scene and negotiate entrance to the wreckage before the victim can be treated. As per the addition operation, the new learning step is inserted into the scenario plan. The planner is invoked, and the learning step is decomposed into three world steps. In order to satisfy a precondition of 'Contact Guard', a new step, 'Restrict Area', is added from the operator library. Because, the world steps in 'Negotiate Safe Passage' are still dead end steps the link from the previous 'Find Victim' step to 'Cleanse Wounds' is replaced with a link from 'Escort to Victim'. Because the 'Find Victim' step is a dead end step and no longer required, it is removed.

4. Adapted Scenarios for Training Games and Simulations

As depicted in Figure 2, the adapted scenario is executed in a game environment to expose the learner to the scenario content. Our technical approach enables us to adapt scenarios for different game engines by swapping planning domains. We are currently

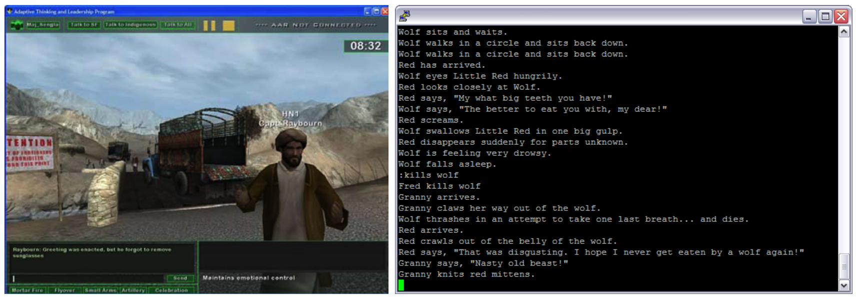

Figure 7. Screen captures of ATL [10] (left) and Automated Story Director in a MOO [7] (right), target platforms for the Scenario Adaptor.

targeting two game engines. The first is a multiplayer online game with human role-players. The second is a single-player game with NPCs.

In multiplayer online environments such as ATL [10] and DARWARS Ambush NK! [11], human readable scripts may be generated for the instructors and roleplayers. Narrative discourse planning [17,18] is being employed to express the scenario in human readable text. The scenario plan is input to a discourse planner customized for the purpose of generating scripts that can be read and enacted on by human role players. For multiplayer online games, enemy and neutral characters are played by trained role-players. The text is suggestive instead of providing exact dialogue to enable role-players to improvise and create a more seamless learning experience.

Because it is not always possible to have trained role-players to support the learner in his or her learning experiences, we also target single-player games in which enemy and neutral characters are controlled by intelligent agents. As proof-of-concept, our game engine is a MOO (a textual multiplayer environment) populated by semi-autonomous character agents. This is a technique used in the Automated Story Director drama management system [7], except we have replaced the "drama manager" with a script execution system that sends instructions to character agents to execute. In this approach, instructions are high-level descriptions of goals the characters should achieve. The agents decompose the instructions into primitive behaviors such as movement, speech (hard-coded lines), and gestures using a reactive planner. The semi-autonomous character agents are assumed to have a wide repertoire of behaviors that they can execute that correspond to the actions that can be included in a scenario from the Scenario Adaptor. Further description of the agents is beyond the scope of this but technical details of agent execution can be found in [7].

5. Conclusions

The adaptation of scenarios may make more more effective computer-based learning environments by creating more efficient training experiences. Individualization of scenarios has the potential to improve engagement. Learners may spend less time on tasks that are either too easy or too difficult and more time on tasks that are challenging but within reach. New algorithms are needed to adapt scenarios, and future work will finalize these algorithms and test learning outcomes.

References

[1] Mark O. Riedl and R. Michael Young. An intent-driven planner for multi-agent story generation. *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, New York*, 2004.

[2] James R. Meehan. Tale-spin, an interactive program that writes stories. In *Fifth International Conference on Artificial Intelligence (IJCAI 77)*, 1977.

[3] Michael Lebowitz. Story-telling as planning and learning. *Poetics*, 14, 1985.

[4] Ruth Aylett Sandy Louchart Joao Dias and Anna Paiva. Fearnot! an experiment in emergent narrative. In *Proceedings of the 5th International Conference on Intelligent Virtual Agents*, 2005.

[5] Marc Cavazza, Fred Charles, and Steven J. Mead. Planning characters' behaviour in interactive storytelling. *Journal of Visualization and Computer Animation*, 13:121–131, 2002.

[6] R. Michael Young, Mark O. Riedl, Mark Branly, R.J. Martin, and C.J. Saretto. An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development*, 1, 2004.

[7] Mark O. Riedl, Andrew Stern, Don Dini, and Jason Alderman. Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning*, 4(2), 2008.

[8] Bradford Mott, Sunyoung Lee, and James Lester. Probablistic goal recognition in interactive narrative environments. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 187–192, Boston, Massachusetts, 2006.

[9] Brian Magerko, Bob Wray, Lisa Holt, and Brian Stensrud. Improving interactive training through individualized content and increased engagement. In *Proceedings of I/ITSEC 2005*, Orlando, FL, 2005.

[10] Elaine M. Raybourn, Ed Deagle, Kip Mendini, and Jerry Heneghan. Adaptive thinking & leadership simulation game training for special forces officers. In *I/ITSEC 2005 Proceedings, Interservice/ Industry Training, Simulation and Education Conference Proceedings*, Orlando, Florida, USA, November 2005.

[11] Elaine M. Raybourn, Bruce Roberts, David Diller, and Leslie Dubow. Honing intercultural engagement skills for stability operations with darwars ambush! nk game-based training. In *Proceedings of the 26th Army Science Conference*, 2008.

[12] John E. Deaton, Charles Barba, Tom Santarelli, Larry Rosenzweig, Vance Souders, Chris McCollum, Jason Seip, Bruce W. Knerr, and Michael J. Singer. Virtual environment cultural training for operational readiness (vector). *Virtual Real.*, 8(3):156–167, 2005.

[13] H. Chad Lane, Mark Core, David Gomboc, Arshish Karnavat, and Milton Rosenberg. Intelligent tutoring for interpersonal and intercultural skills. *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC 2007)*, 2007.

[14] Judy Kay. Lifelong learner modeling for lifelong personalized pervasive learning. *IEEE Transactions on Learning Technologies*, 1(4), 2008.

[15] H. Chad Lane, Mark Core, Dave Gomboc, Mick Birch, John Hart, and Milton Rosenberg. Using written and behavioral data to detect evidence of continuous learning. In *Proceedings of the UMAP 2009 Workshop on Lifelong User Modeling*, 2009.

[16] R. Michael Young and Martha E. Pollack. Decomposition and causality in partial-order planning. In *Proceedings of the Second International Conference on Artificial Intelligence and Planning Systems, Chicago, IL*, 1994.

[17] R. Michael Young and Johanna D. Moore. Dpocl: A principled approach to discourse planning. In *Proceedings of the Seventh International Workshop on Text Generation*, Kennebunkport, ME, 1994.

[18] Mark O. Riedl. *Story Generation: Balancing Plot and Character*. PhD thesis, North Carolina State University, September 2004.