Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training

Mark O. Riedl [1], Andrew Stern [2], Don M. Dini [1], and Jason M. Alderman [3]

[1] Institute for Creative Technologies, University of Southern California
Marina Del Rey, California USA
[2] Procedural Arts LLC, Portland, Oregon USA
[3] School of Literature, Communication, and Culture, Georgia Institute of Technology
Atlanta, Georgia USA
Email: riedl@ict.usc.edu; andrew@proceduralarts.com; dini@ict.usc.edu; jasonald@gatech.edu

Abstract: Modern computer systems have the ability to make the storytelling experience interactive by involving a participant or learner as a character in the narrative itself. We present a framework for creating interactive narratives for entertainment, educational, and training purposes based on a type of agent called an *experience manager*. An experience manager (a generalization of a *drama manager*) is an intelligent computer agent that manipulates a virtual world to coerce a participant's experience to conform to a set of provided properties. Our realization of the experience manager automatically generates narrative content in order to adapt to the user's actions in the virtual world. The experience management framework has been used to develop an interactive version of Little Red Riding Hood and an interactive practice environment called IN-TALE for educating and training cognitive skills such as situation awareness, cultural awareness, leadership, and decision-making.

Keywords: Interactive narrative; Storytelling in education and training; Drama management; Experience management; Intelligent tutoring systems

1. Introduction

Storytelling is a modality for communicating: it is a technique for entertaining, educating, and training. Narrative is most often considered an artifact that is created by a human for a particular purpose. However, it is important not to overlook the fact that narratives are meant to be experienced by an audience (a reader, listener, watcher, interactor, etc.). Researchers have begun to investigate narrative from the perspective of the audience, looking at the cognitive processes that are invoked by readers of narratives [17] [19] [20]. This has led some (e.g., [11] [18]) to conclude that narrative is a mental construct or a mode of thought.

Recently, there has been growing interest in stories as artifacts that can augment interactive entertainment (e.g., computer games) products such as video games. This interest in storytelling in interactive entertainment has lead to the development of the notion of an *interactive narrative*. An interactive narrative is an approach to interactive entertainment in which a system attempts to tell a story to an interactive participant. In order to distinguish interactive narrative systems from other types of interactive entertainment, an interactive narrative allows the user to make decisions that directly affect the direction and/or outcome of the story being told by the system.

The goal of an interactive narrative system is to balance the seemingly competing requirements for narrative coherence and perceived user self-agency [44]. Narrative coherence is the idea that the events that occur in a narrative have meaning and relevance to each other and to the outcome. Perceived user self-agency is the idea that the user, while immersed in a virtual world, perceives his or herself to be capable of making meaningful decisions. The Choose-Your-Own-Adventure novels, originally published by Bantam Books, are canonical examples of interactive narratives that balance narrative coherence and user self-agency by separating the experience into non-interactive narrative interspersed with decision-points.

We take the perspective that narrative is a cognitive phenomenon and, accordingly, adopt an approach to interactive narrative as the management of a virtual world as a way to increase the likelihood that a human participant will perceive his or her experience in a virtual world as narrative. We explore the implications of this perspective on the problem in the next sections, building towards an argument for a particular approach to interactive narrative systems called *experience management.* Our experience management approach is realized in a framework that has been used to develop two prototype examples, demonstrating the capability in entertainment and educational contexts.

1.1 Reconstructed Narrative

We understand the world we are situated in and our progression through that world as a reconstructed narrative (sometimes also referred to as "storification" [3]). That is, we observe events in the world and attempt to mentally fabricate understanding by selectively arranging events into narratives. We understand the world by telling ourselves stories about how we have changed the world and witnessed the world change. Our understanding of the world is achieved by "constructing reality" as a sequence of related events from our senses [12]. Whereas we tend to understand inanimate objects through cause and effect, we attempt to understand the intentional behavior of others through a sophisticated process of interpretation with narrative at its core [11]. There is evidence that suggests that we, as humans, build cognitive structures that represent the real events in our lives using models similar to the ones used for narrative in order to better understand the world around us [11]. This *narrative intelligence* [8] [31] is central in the cognitive processes that we employ across a range of experiences, from entertainment contexts to active learning.

Given the propensity of the human mind to construct narrative from experience, an emergent approach to interactive narrative has appeal. Emergent narrative [2] [3] is a particular AI approach to creating interactive stories in which a user interacts with other people or "virtual humans" – anthropomorphic entities that are controlled by computer systems (c.f., [13] [41] [53]) – in a virtual world. The story, from the perspective of an outside observer, is the culmination of events (e.g., the behaviors and dialogue of the user and the synthetic humans) that occur during the simulation. Emergent narrative is particularly meaningful for virtual worlds in that participation in a virtual world – including computer game worlds – can be regarded as participating in an unfolding narrative. That is, conceptually narrative can be constructed in the mind of the user at the same time it is being created through simulation.

Virtual world systems with purely emergent approaches to narrative have met with varying degrees of success. Aylett [2] notes that narratives do not always emerge that have recognizable structure or coherent, meaningful outcomes. What does this mean for virtual worlds? For virtual worlds that aim to provide entertaining experiences, the lack of a satisfying story-like narrative can be disappointing. Likewise, simulations and other constructive virtual environments with training and educational goals cannot guarantee experiences that reinforce knowledge acquisition. Consequently, games and training systems often turn to some form of guidance of user experience.

Story is particularly important for games. Aside from a few simulation-based games, most adventure games and first-person shooters incorporate story into their systems by constraining the virtual world to force the player's experience to conform to closely controlled sequences of events. This constraining function is often spatial in nature, i.e., the movement through a virtual environment constitutes progression through a story [24]. There are other models of guided experience. In the next section we describe one approach to guiding a user's experience to influence the user's cognitive narrative reconstruction

1.2 Drama Management

Interactive narratives place primary importance on the user's experience of story. There are many ways to achieve an interactive narrative. Mateas and Stern [32] describe a continuum of technical approaches ranging from *strong autonomy* to *strong story*. The strong autonomy approach advocates that interactive narrative experiences be created procedurally by simulating a virtual environment populated by autonomous agents that play the roles of characters. Emergent narrative systems fall under the strong autonomy approach. Examples of strong autonomy systems are [14] and [4]. The strong story approach advocates that a singular decision-making process that operates as if it were a hypothetical author, choosing the activities of all story world characters in a centrally coordinated fashion, generate interactive narratives. Examples of strong story systems are [5], [28], [33], [36], [37], [44], [49], [50], [54], [59], and [61].

Many interactive narrative systems – especially those that would be classified as strong story approaches – utilize an intelligent agent called a *drama manager*. A drama manager [25] is an agent that attempts to coerce a virtual world so that a player's interactive experience to conform to some pre-existing aesthetic.

From the participant's perspective, narrative is reconstructed from their interactions with the virtual world and the computer-controlled characters that are also immersed in the virtual world. The drama manager attempts to guide the participant's experience without over-constraining the participant to the point where he or she feels he or she has no self-agency. By balancing narrative considerations against participant self-agency, the resulting experience is something that is neither wholly dictated by the experience manager nor the participant.

In the absence of computational models of aesthetics, drama manager often utilize pre-scripted narrative-like data structures such as plot graphs or branching narrative sequences. A plot graph [25] [59] is a data structure that defines a set of partially ordered story elements from which the drama manager chooses from to assemble the story in real-time. A branching narrative [10] [48] is directed graph that lays out all possible narrative sequences that can occur based on user decisions at pre-determine choice-points. A non-exhaustive list of systems that can be described as drama managers includes [5], [28], [33], [36], [37], [49], [50], [54], [59], and [61].

1.3 Experience Management

The implied goal of a drama management system is to entertain. However, recent research has suggested that interactive narratives can be used for education and training (c.f., [4] [29] [30]). We submit that a drama manager is an instance of a more general type of system that we refer to as an *experience manager*. An experience manager is an agent that attempts to coerce a virtual world so that a user's interactive experience conforms to a set of provided properties. An experience manager whose given properties described dramatic attributes of story would be referred to as a drama manager, whereas an experience manager who's given properties described pedagogical attributes would not be described as a drama manager.

The concept of experience management, while new to the field of interactive narrative and interactive entertainment (i.e., computer games), is similar in many ways to concepts from the field of intelligent tutoring for some time. Intelligent Tutoring Systems (ITS) attempt to reproduce with computer systems the learning gains in students who are tutored by human tutors [51] [58]. The general behavior of an ITS can be defined by a process with two nested loops [56]. The outer loop is one of task selection. The inner loop traces the student's progress through a task. The outer loop is a form of experience management because the ITS is decided reactively based on the student's past performance, what task the student should be exposed to next in order to scaffold and support the student's construction of knowledge. One might refer to the outer loop of an ITS as a form of experience management. An excellent example of tutoring as experience management is described in [35], although the authors do not use the term "experience management."

Experience management systems, however, are not necessarily intelligent tutoring systems. One of the prototype systems described in this article – the IN-TALE system – demonstrates experience management for education and training. However, we do not consider IN-TALE as an ITS because it provides an adaptive practice environment and does not provide hints, give feedback, or attempt to remediate incorrect understanding of concepts.

Experience management as a technique for interactive narrative emphasizes the reconstructed narrative experience of the user, whether for entertainment, education, or training. We believe that experience management based approaches to interactive narrative hold potential for computer-based education and training in virtual worlds. Baumeister and Newman [7] describe a phenomenon where tacit expert knowledge can only be accessed by experts through autobiographical narratives in which the knowledge was applied. Likewise narrative case studies are a good technique for transferring knowledge from experts to novices [52]. The use of stories for tacit knowledge transfer is justification for story-based case method training [23]. It is also justification for more interactive storytelling environments in which learning by doing [1], especially when merged with narrative accounts of best practices.

In this paper, we present a framework for producing interactive narrative experiences with wide ranging applicability, from entertainment to education and training. As an experience manager, our framework is a *strong story* approach, although we incorporate semi-autonomous agents into the framework resulting in some attributes similar to *strong autonomy* systems. Our interest in pedagogical as well as entertainment applications of interactive narrative suggest a framework based on an experience manager that can be readily adapted to dramatically- and pedagogically-inspired narratives.

The goal of our experience management framework is to enable the creation interactive narratives that afford the participant a high degree of self-agency in a virtual world while simultaneously delivering a coherent narrative experience centered on aesthetic or pedagogical attributes. It is possible for the participant to perform actions in the virtual world that directly conflict with the system's ability to tell the intended narrative [44]. When this happens, the framework invokes a special agent, called the *Automated Story Director*, to dynamically adapt the narrative structure in order to balance the goals of the system (to tell a narrative with given attributes) without interfering with the user's self-agency. A *generative* experience management system does not require the pre-specification of all possible narratives that can occur.

Our experience management framework is demonstrated by two prototype applications that instantiate the framework. One prototype, described in this article, implements an interactive world based on a version of the Little Red Riding Hood story. The Little Red Riding Hood interactive story demonstrates the generative experience management framework in the context of an entertainment-based application. The other prototype described in this article, the Interactive Narrative Tacit Adaptive Leader Experience (IN-TALE) [45] [46], is a military training system designed to enable learners to interactively practice skills such as leadership, situation awareness, cultural awareness, decision-making, and other cognitive skills. Both the Little Red Riding Hood and IN-TALE prototypes enable a user to interact in a virtual world by controlling an avatar and assuming the role of a story world character. The systems attempt to tell a narrative in which the user is an integral participant. In the case of the Little Red Riding Hood prototype, the intended narrative is one that resembles the original story. In the case of IN-TALE, the intended narrative is one that requires the learner to be exposed to certain conditions in which learning can occur and skills can be practiced.

In the following sections we describe our framework for interactive narratives. We begin with a description of the central component to the framework – the generative experience manager called the Automated Story Director – in Section 2. In Section 3, we describe the framework that is built around the Automated Story Director. In particular, we emphasize how semi-autonomous character agents are incorporated into the framework to work in coordination with the Automated Story Director. Authoring is also discussed here. To date, the framework has been used to produce two prototype systems. One prototype, described in Section 4, is an interactive narrative based on Little Red Riding Hood. The other prototype, the Interactive Narrative Tacit Adaptive Leader Experience (IN-TALE), described in detail in Section 5, is an educational interactive narrative system designed to be a practice environment for training military cognitive skills such as situational and cultural awareness. We conclude the paper with discussions of related work (Section 6), limitations and future work (Section 7), and general conclusions (Section 8).

2. Automated Story Director

Our experience manager – the *Automated Story Director* – is responsible for coercing the participant's experience to attempt to bring about certain, given properties. The Automated Story Director takes as one of its initialization parameters a narrative called the *exemplar narrative*; it is a narrative that the system expects that the participant will experience if he or she were to do everything that he or she is expected to do. The exemplar narrative encodes either implicitly or explicitly all the desired properties of the participants experience, whether dramatic or pedagogical.

One of the primary challenges of interactive narrative systems is handling the interactivity afforded the participant in an interactive narrative. It is possible that the participant will not perform all the actions or make all the decisions that are expected of him or her by the exemplar narrative. Indeed, it is possible that the participant may perform actions or make

```
(operator Plant-Bomb
  :parameters (?character ?bomb)
  :precondition ((has ?character ?bomb)
                 (:not (detained ?character))
  :effect ((armed ?bomb) (planted ?bomb)
          (:not (has ?character ?bomb))))

(operator Acquire-Object
  :parameters (?acquirer ?object ?owner)
  :precondition ((has ?owner ?object)
                 (:not (detained ?acquirer))
                 (:not (detained ?owner)))
  :effect ((has ?acquirer ?object)
          (:not (has ?owner ?object))))
```

Fig. 1. Example planning operators.

decisions – intentionally or inadvertently – that directly contradict or invalidate the exemplar narrative. Due to this possibility, the experience manager must be able to adapt the exemplar narrative to preserve the participant's perception of self-agency, narrative coherence, and the given desired properties of the participant's experience (dramatic and/or pedagogical). This means that the participant is allowed to do what he or she wants and that the system will adapt itself to that, thus preserving the participants perception that they are in control of their own experience. Adaptation is performed in such a way that certain desired properties of the narrative artifact (not the participant's perspective of the narrative nature of their emergent experience) are preserved to the extent possible.

In the next section we describe how narrative structures such as the exemplar narrative are encoded in the Automated Story Director. The computational representation allows the Automated Story Director to operate on narrative structures and to generate new narrative structures without intervention by a human author. Section 2.2 extends these concepts and describes how a system designer/author can exert his or her authorial intent on narrative content automatically generated by the system. The remainder of the section describes operation of the algorithms for generating and executing narrative interactively. Section 2.3 describes how the system generates new narrative content. Section 2.4 describes how narrative content is executed. Section 2.5 describes how the user is monitored and how the system decides when to adapt.

2.1 Computational Representation of Narrative

In order to adapt a narrative, the narrative must be represented in a form that lends itself to manipulation by artificial intelligence reasoning algorithms. Following [44] [61], we represent narratives as partially-ordered plans. A plan contains steps – events that change the state of the world – and annotations that explicitly mark the temporal and causal relationships between all steps in the plan, defining a partial order indicating the steps' order of execution [57].

Plan steps, which represent plot points, are represented in a STRIPS-like language. Figure 1 shows some un-instantiated operators from a domain operator library that provides a source of basic knowledge about actions that story world characters can perform and how those actions effect the story world. The operator parameters are variables that will be bound as appropriate to such things as character or object instances. Preconditions are conditions in the world that must be true for the operator to be applicable. Effects are conditions in the world that become true after successful execution of the instantiated operator.

The first operator in Figure 1 describes an action where a character plants a bomb. Preconditions state that that character must have the bomb in his or her possession and not be detained. The effects of the operator are that the bomb is planted and armed and that the character no longer has the bomb in his or her possession. The second operator in Figure 1 describes an action whereby one character acquires an object from another character. Preconditions state that one character, the owner has the object and that both characters are not detained. The effects of the operator are that the acquiring character has the object and the original owner no longer has the object.

A plan is made up of a set of instantiated operators, meaning the parameters have all been bound. Temporal and causal annotations describe constraints on the order in which instantiated operators – called plan steps – can be executed. One type of annotation, called a *temporal link*, is used to mark an ordering constraint between steps in the plan. A temporal link constrains one step to necessarily be executed before another step. Another type of annotation, called a *causal link*, is used to mark a causal relationship between steps in the plan. In a plan, a causal link relates the effect of one plan step to a precondition of another plan step that is temporally constrained to occur later than the first operator. Together, temporal and causal links determine the execution order of steps in a plan. The ordering can be partial, meaning there can be steps that are not temporally constrained relative to each other and can thus be executed in any order or concurrently.

Temporal and causal links are created automatically during the planning process. A plan can only be considered to be sound and complete when every precondition of every step in the plan is satisfied by a causal link and the temporal ordering prevents the effects of any one step from negating the preconditions of another step [38].

Figure 2 shows an example of a plan (from the Little Red Riding Hood prototype described in Section 4). Gray boxes represent plan steps. Solid arrows are causal links – the label on the links describe the condition that is established by the originating step that satisfies a precondition of the terminus step. Dashed arrows represent temporal constraints. If two plan steps are not strictly ordered by temporal or causal links, those steps can be executed in any order or simultaneously (if possible) [26]. Not all causal and links are shown.

Causal dependency planning [57] operates in a backward chaining fashion as a process of flaw repair. A flaw is an annotation on an incomplete plan that specifies how the plan will fail to execute. One type of flaw is an *open condition*, where a plan step has a precondition that is not causally satisfied by a preceding step or the initial world state. Open conditions are repaired by extending a causal link from a preceding step in the plan that has an effect that unifies with the open condition. If an applicable step does not exist, a new step is instantiated from a library of operators. Further description of the planning process is outside the scope of this article. However, [42] [47] describes a narrative

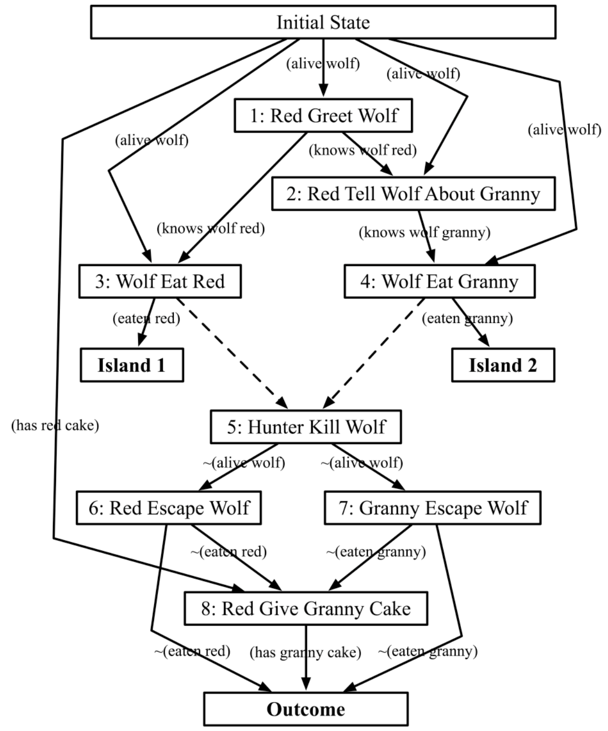


Fig. 2. Example narrative plan set in the Little Red Riding Hood world.

planning algorithm consistent with the generative experience management approach described in this article.

A plan, as a computational representation of narrative, facilitates generation and execution of narratives. However, it is important that the human author be able to exert influence over the types of narratives created by the Automated Story Director. The next section describes an extension to the computational representation that allows the human author to exert their authorial intent on the system.

2.2 Preserving Authorial Intent

The Automated Story Director is a generative approach to experience management, meaning that an automated narrative generation process automatically produces variations on the hand-authored narrative content (e.g. the exemplar narrative). While the human author necessarily cannot have fine-grained control of the participant's experience during runtime, it is a desirable trait for interactive narrative systems – from both dramatic and pedagogical perspectives – to allow the author to be able to constrain the space of possible experiences the participant can have. For an entertainment-based system, the authorial intent may be a set of dramatic circumstances. For an educational or training system, the authorial intent may be to introduce the user to situations relevant to the course of study.

The Automated Story Director provides a mechanism for encoding authorial intent based on special data structures called *islands*. Islands – a term coined to refer to a technique for controlling the form of solutions generated by planners [22] – are intermediate states in a search space, through which all solutions to a planning problem must pass. Islands inform the planner as to what valid solution plans should look like, conceptually speeding up the planning process. Plans that do not satisfy each island state description at some point between the initial state and the end state are effectively pruned. We use islands as a way for the human user to inject guidance into the narrative generation process used by the experience manager.

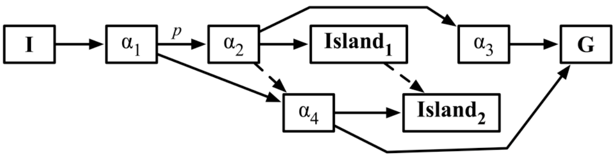
In our computational representation of narrative, islands are implemented as a special type of plan step that have preconditions describing the intermediate world state but no effects. Islands are provided at the time of planner initialization and describe world states that must be achieved at some intermediate time during plan execution. If more than one island is given, there can be pre-specified temporal links between them so that islands must occur in the resulting, complete plan in a particular order. In this way, the existence of islands constrains the space of plans that can be searched by the planner. That is, the planner cannot consider any plan in which the world state described by an island will not be achieved during plan execution. Unlike total-order state-space search algorithms, causal dependency planners (e.g., [57], [60], and [42] [47]) search through the space of partial plans [57]. When islands are used, the planner is instantiated with *virtual* plan steps – that is, plan steps that are not executed – inside the initial, usually empty, plan.

For example, in an implementation of an interactive Little Red Riding Hood story, islands may specify that valid narratives are those in which (a) Little Red Riding Hood is at some point in the state of being eaten (island 1 in Figure 2), and (b) Little Red Riding Hood is at some later point in the state of not being eaten (part of the Outcome in Figure 2). This prevents the experience manager from considering any narrative plans in which Little Red Riding Hood is not eaten at some point, or in which Little Red Riding Hood is eaten but not rescued. The goal – or outcome – of a plan can be considered a special type of island. Once authorial intent is encoded into the exemplar narrative plan, the system can reason about how to adapt that narrative in order to handle user interactivity.

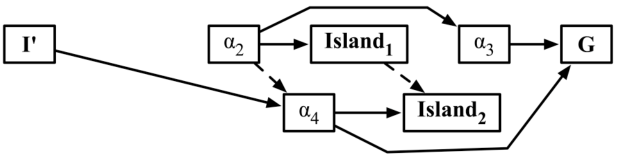
2.3 Anticipating Necessary Narrative Plan Adaptations

Using planning structures to model scenarios is advantageous because a plan can be analyzed for points in which failure can occur due to unpredictable and interactive behaviors performed by the participant. We use a technique similar to that described in [44] to analyze the causal structure of the scenario to determine all possible inconsistencies between plan and virtual world state that can occur during the entire duration of the narrative. Inconsistencies arise due to participant actions that change the world. For every possible inconsistency that can arise that threatens a causal link in the plan, an alternative narrative plan is generated. Unlike [44], we do not attempt to prevent the user from causing undesirable world state changes. Instead, we use a tiered replanning approach. For each potential inconsistency that can arise, the following repair processes are tried in order until one succeeds in repairing the narrative:

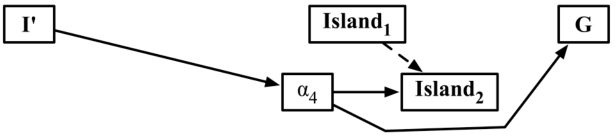
(i) The threatened causal link is removed, an open condition flaw annotates the precondition on the plan step that was previously established by the removed causal link, and the planner is invoked.



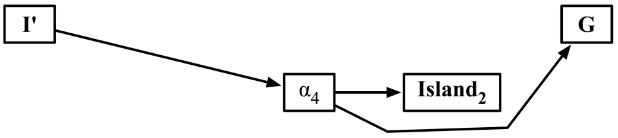
(3.a.) A hypothetical narrative plan.



(3.b.) Strategy (i) after pre-processing.



(3.c.) Strategy (ii) after pre-processing.



(3.d.) Strategy (iii) after pre-processing.

(3.e.) Strategy (iv) after pre-processing.

Fig. 3. Illustration of the tiered re-planning strategies used by the Automated Story Director.

(ii) The threatened causal link is removed, the plan step that was at the terminus of the causal link and all other plan steps (except islands) that are causally dependent (e.g. there is a path from the threatened causal link to a given step through the directed graph made up of plan steps and causal links) are removed, open conditions flaw annotations on the remaining steps are created as necessary, and the planner is invoked.

(iii) The threatened causal link is removed, the plan step that was at the terminus of the causal link and all other plan steps (including islands) that are causally dependent are removed, open conditions flaw annotations on the remaining steps are created as necessary, and the planner is invoked.

(iv) The remaining plan is discarded, a new set of goals and islands is selected with open condition flaw annotations as necessary, and the planner is invoked.

The tier (iv) strategy requires the user to provide, *a priori*, multiple sets of goals and islands ordered according to the desirability that narrative plans satisfy those goals and islands. If the final tier of replanning fails, we resort to a non-managed virtual world, relying completely on the autonomy of computer-controlled characters to create an emergent narrative experience.

2.3.1 Replanning Example

To illustrate the tiers of repair strategies, consider the narrative plan in Figure 3.a. The plan contains for steps, α_1 through α_4, and two islands. *I* and *G* are the initial state and goal state, respectively. Aside from the causal links, α_2 is temporally ordered before α_4 and $island_1$ is temporally ordered before $island_2$. Step α_1 establishes condition *p* in the world, which is necessary for step α_2. This relationship is captured by a causal link annotation. Suppose the Automated Story Director is considering the possibility that the user performs an action in the world that causes $\sim p$ to become true during the interval between α_1 and α_2.

The tier (i) re-planning strategy is invoked first. A copy of the plan is made and updated to reflect the state of the plan after all steps preceding the interval in question are executed. That is, the initial state now represents the world state after α_1 is executed, and step α_1 is removed. The tier (i) strategy removes only the causal links in the interval in question that are threatened by the possible user action. Figure 3.b. shows the copy of the plan after tier (i) pre-processing. The plan in Figure 4.b. is flawed because step α_2 has preconditions that are not satisfied by any causal links. The planner is invoked to repair the plan, which is now flawed.

Suppose that the tier (i) strategy fails. That is, the planner cannot find any partially ordered sequence of new steps that can fill the gap. If the re-planner fails, the Automated Story Director advances to the tier (ii) re-planning strategy. The tier (ii) strategy removes the causal link threatened by the potential user action, the steps immediately succeeding the interval that are satisfied by the threatened links (e.g. step α_2), and all *causally downstream* steps except islands. A causally downstream step is any step α_i such that there is a path from a removed step to α_i. In this example, step α_3 is causally downstream. Note that although α_4 is temporally constrained to occur after α_2, it is not in fact causally dependent on α_2 and therefore is *not* removed. Figure 3.c. shows the copy of the plan after tier (ii) pre-processing. The planner is then invoked to repair the plan.

Suppose that tier (ii) strategy fails. The Automated Story Director advances to the tier (iii) re-planning strategy. The tier (iii) strategy is similar to tier (ii) except that causally downstream steps are removed *including islands*. The assumption is that tier (ii) failed because the islands could not be achieved. Figure 3.d. shows the copy of the plan after tier (iii) pre-processing. The planner is then invoked to repair the plan.

Finally, suppose that the tier (iii) strategy fails. The Automated Story Director advances to tier (iv). The tier (iv) strategy deletes the entire plan, replaces the goal state *G* with a new goal state G', and instantiates any number of new islands. The new goal and islands come from a user-specified list of alternative goals. Figure 3.e. shows the plan after tier (iv) pre-processing. In this example, there is only one new island.

2.3.2 Offline Computation of Contingencies

Narrative replanning is performed offline to avoid delays due to computation [44]. The result of this process is a tree of contingency plans in which each plan represents a complete narrative starting at the initial world state or at the point in which an inconsistency can occur. If the user performs an action that causes an inconsistency that threatens the narrative plan, the system looks up the appropriate branch in the tree of contingencies and seamlessly begins directing the believable agents based on the new narrative plan. User actions that do not threaten causal links are considered by the Automated Story Director to be consistent with the current narrative structure [44].

Figures 5 and 8 show portions of the trees automatically generated for Little Red Riding Hood and IN-TALE, respectively. As before, inside the plan nodes solid arrows represent causal links and dashed arrows represent temporal links. The vertical i-beams alongside each plan node represent intervals in which the participant has the opportunity to create an inconsistency between the virtual world and the plan. The arrows *between* plan nodes indicate which contingency narrative plan should be used if an inconsistency does in fact occur during interactive execution. Note that the tree of contingency plans can be potentially infinite in depth. As a matter of practicality, we cap the depth to which the tree can grow. We use a simple user model to inform the Automated Story Director about which inconsistencies for any given narrative plan are more probable, and focus on making those parts of the tree more complete.

The contingency tree is necessary for dynamic execution. By pre-generating the tree, the Automated Story Director can rapidly switch to alternative narrative plans when user actions make this necessary. This can be done without being held up by planner computation time, which can be unpredictable.

2.4 Narrative Plan Execution

For narrative plan execution, plan steps in the current narrative plan are interpreted as abstract event descriptions at the level commonly associated with plot. A plot is an outline of main situations and events in a narrative [40]. Originally, the current narrative plan is the exemplar narrative, although the current narrative plan can change. The plan steps and their causal relationships are used to generate directives to computer-controlled, semi-autonomous agents playing the roles of story world characters (see Section 3.2). It is through directives to the story world characters that the Automated Story Director manipulates the virtual world and consequently the participant's experience. Directives to story world characters take the following forms:

- *Prescriptive directive*: Direction to achieve some world state that is desirable to the experience manager and moves the plot forward. Prescriptive directives should be of sufficiently high level that character agents are free to achieve the directive in a way that best demonstrates the characteristics of that character.
- *Proscriptive directive*: Direction to avoid particular states in the virtual world that are contradictory to the intermediate requirements of the experience manager's plot representation.

The drama manager must derive directives from its internal plot representation. Prescriptive directives are derived directly from the plan steps. That is, a plan step specifies a world state that must be achieved. The Automated Story Director determines which story character is responsible for achieving this world state and redefines the plan step as a goal for the character agent to achieve. It is often the case that the character responsible for achieving the step is encoded directly into the step as the entity that is changed or creates the change.

Proscriptive directives are necessary because the computer-controlled characters we use have some degree of autonomy (See Section 3) and are therefore capable of performing actions that interfere (e.g. create inconsistencies) with the narrative plan. Proscriptive directives are derived from the causal relationships between plan steps. Specifically, for every plan step in the drama manager's representation of expected user experience, there are certain world state conditions that must be maintained. These conditions are established by plan steps that occur earlier in the plan structure or by the initial world state. At any given time during dynamic execution of the narrative plan, there exists a cut of the narrative plan (as a directed graph) such that steps that have already been achieved are on one side of the cut and steps that have not yet been achieved are on the other side of the cut. The causal links that cross the cut (e.g. the initiating step is on one side of the cut and the terminus step is on the other side of the cut) represent the conditions that must be maintained. These conditions are formatted into proscriptive directives to the characters, so that they are aware that they should not perform any action that negates a causal link. Interfering with the currently executing narrative plan is a right only bestowed upon the interactive participant.

The set of computer-controlled character agents is the mechanism through which narrative plans are "executed." That is, the Automated Story Director distributes plan steps to computer-controlled agents in the form of prescriptive and proscriptive directives. Computer-controlled character agents treat prescriptive directives as goals. When computer-controlled character agents achieve their directed goals they report back to the Automated Story Director, which then marks the plan step as executed and determines the next step that can be executed.

We find that by separating the system into an experience manager and semi-autonomous character agents, the execution manager can focus on plot-level (high level abstract) details. This allows for a larger repertoire of user actions – including dialogue acts – to be implemented because relatively few user actions threaten the plot when represented at a high level of abstraction. Computer-controlled character agents with a degree of autonomy are capable of providing a large amount of variability in the user's experience, especially with dialogue, without requiring the plot to be adapted. We believe this will invoke in the user a greater sense of local agency. The representation and adaptation of complete plots by the Automated Story Director affords the user global autonomy, although the user may not be aware of it.

2.5 Participant Monitoring and Adaptation

During execution of the system, the participant is an interactive member of the virtual world. As such, the participant is able to perform a wide repertoire of actions, from engaging in discourse with other story world characters to moving around and altering the physical environment. It is possible that some of the actions that the participant may want to perform will change the world or the computer-controlled characters (e.g. mental states, belief states, etc.) in such a way that it becomes inconsistent with the currently executing narrative plan. When this happens, the Automated Story Director must be informed of the changes to the virtual world and react accordingly, adapting the narrative plan by selecting the most applicable contingency plan from the tree of narrative adaptations.

The virtual world is augmented with daemons that monitor the virtual world and the computer-controlled characters for certain world states to occur. For example, one daemon in the Little Red Riding Hood domain would monitor for the death of the Wolf because the premature death of the Wolf significantly impacts the narrative plot progression. If a story world's characters must have certain beliefs for a narrative plan to progress, there must be daemons that inspect the mental state of the computer-controlled characters, since the characters are a fundamental part of the virtual world. Each daemon triggers on one specific world state proposition. The daemons report back truth values for the world state propositions they are monitoring for, but are not otherwise aware of whether the world state change does or does not create an inconsistency with the narrative plan. The daemons report back potential inconsistencies and the Automated Story Director makes the determination of whether a potential inconsistency is an actual inconsistency by comparing the virtual world's state to the current expected state of the currently executing narrative plan.

Suppose the user performs an action in the virtual world that changes the world state in some way. Further suppose that the plan has entered an interval in which that world state is identified as being inconsistent with the plan. If this happens during interactive execution then there is an actual inconsistency (as opposed to a potential inconsistency). There is some plan step somewhere in the immediate or far future of the currently executing narrative plan that can no longer be executed because its preconditions are no longer established. When this occurs, the Automated Story Director looks for a narrative plan in its pre-built tree of contingency narrative plans. The Automated Story Director looks at the tree of narrative plans for the child of the currently executing narrative plan that specifically repairs the inconsistency. This selected child plan is a candidate to become the new currently executed narrative plan.

It is possible, however, that other potential inconsistencies have occurred and were recorded previously that create actual inconsistencies with the new candidate narrative plan. These previous world state changes may not have threatened the previous narrative plan but now affect the candidate child plan. That is, the world state is different from the initial state that was expected by the candidate plan in a way that has implications for causal links in the plan. If this is the case, the Automated Story Director looks to the children of the candidate plan, and the process repeats until a candidate plan is found that is not threatened by any potential inconsistencies recorded by the Automated Story Director.

3. The Experience Management Framework

The framework for building interactive narratives consists of three components: a virtual world, the Automated Story Director, and a set of semi-autonomous character agents. The virtual world is a basic necessity. The virtual world and semi-autonomous character agents are described below. While the virtual world is a necessary basic ingredient in any interactive narrative, the primary way that the Automated Story Director manipulates the virtual world is through the semi-autonomous character agents.

3.1. The Virtual World

The virtual world is a simulation environment of arbitrary complexity and realism. The framework does not specify any particular virtual world technology be used but it must support the following features:

(i) The virtual world must be interactive and the participant interacts with the virtual world through an embodied avatar;
(ii) The virtual world must enable computer-controlled character agents to interact with the virtual world through embodied avatars; and
(iii) The virtual world must support an API that enables the Automated Story Director (through daemons) to monitor the participant and the state of the world.
(iv) The virtual world can support an API through which the Automated Story Director directly alters the world state (i.e., not through the actions of computer-controlled character agents).

Currently, none of the example applications implemented in the generative experience management framework use feature (iv). However, [21] describes situations in which an omnipotent director agent would want to make changes to the virtual world (e.g., locking a door), and also notes the conditions under which changing the world state are favorable (e.g., the user doesn't know whether the door is locked or unlocked).

3.2 Semi-Autonomous Character Agents

Autonomous, embodied agents – also referred to as virtual humans – inhabit virtual worlds by controlling 3D anthropomorphic avatars [41] [53] [13] and provide a rich modality for human-computer interaction. Unlike other forms of artificial intelligence designed to solve one particular problem, virtual humans employ a variety of artificial intelligence systems for situated task performance and understanding and producing natural language and body language.

Any social interaction between humans or between humans and artificial characters can give rise to emergent narrative. Thus many researchers believe the goal in creating virtual humans should be to achieve a level of *believability*.

Believability does not refer to the trustworthiness of an artificial intelligence agent. Rather, a believable agent is one that possesses behavior traits that facilitate one's suspension of disbelief that the agent is a real person [6]. For an enumeration of many of the traits desirable in a believable agent, see [27]. Believable agents are typically autonomous, meaning they are capable of choosing their own goals and choosing how to execute those goals. Autonomy affords reactivity, which can be beneficial for providing a rich modality for interaction with a human participant because they can respond to dynamic and unpredictable actions of a human participant.

To provide a rich interactive and social modality for a human participant that is also guided by an experience manager, we employ *semi-autonomous character agents*. A semi-autonomous character agent is a believable agent that is capable of acting autonomously at times and also receiving directions from an experience manager at other times. The notion of believable agents with some degree of autonomy being directed by an external source is common in drama management/experience management [9] [25] [32]. For example, Blumberg and Galyean [9] describe a way of controlling the behaviors of autonomous virtual agents through *prescriptive* – "do X" – and *proscriptive* – "I don't care what you do as long as you don't do Y" – directions. Most drama management systems to date use prescriptive directions character agents only because the agents have been programmed or instantiated to innately never perform actions that are inconsistent with the drama manager's plot.

We employ semi-autonomous character agents that are reactive and appear intelligent, motivated, and reactive. Like other believable agent technologies, we place an emphasis on the appearance of intelligence – referred to as a "broad but shallow" approach [6]. Our agents are partly composed of a broad, general collection of local autonomous behaviors that are designed to afford suspension of disbelief. Local autonomous behaviors (LABs) such as working, running errands, shopping, etc. supply agents with a "rich inner life." That is, agents can perform a wide repertoire of behaviors in a convincing manner but without performing "deep" reasoning. It is important that NPCs are capable of acting to bring about a specific narrative. Narrative-specific interactive events such as confronting the player and acquiring, planting, and detonating an explosive device, are carried out by narrative directive behaviors (NDBs). Narrative directive behaviors are incorporated into the agents' behavior repertoires before run-time and triggered by high-level narrative direction from the Automated Story Director.

Agent Behavior Modeling. To achieve the desired life-like qualities we implemented our agents using the reactive planning language ABL (A Behavior Language) [34]. The ABL language was initially created for the interactive drama, *Façade* [32] [33] and is designed to support the detailed expression of artistically-chosen personality, automatic control of real-time interactive animation, and architectural support for many of the requirements of believable agents. However, our agent architecture differs from that used in *Façade*. In our agents, there are two broad categories of agent behaviors. Local autonomous behaviors (LABs) are the somewhat generic, re-usable "inner life" activities such as working, running errands, shopping, etc. Narrative directive behaviors (NDBs) are scenario-specific and are triggered by the Automated Scenario Director.

Local Autonomous Behaviors. Local autonomous behaviors (LABs) are implemented as loosely structured collections of sub-behaviors called "LAB goals" that depend on and assert simple events in episodic memory. For example suppose an agent is in the role of a shopkeeper. The *opening the store* LAB may involve the agent *unlocking the store*, *unpacking boxes*, *chatting with assistant*, and *displaying new goods*. Each of these parts is implemented as its own simple LAB goal with ordering constraints between goals. User interactions, should they occur, can easily be inserted during or in between the loosely organized LAB goals. Each individual agent is responsible for selecting and sequencing their local autonomous behaviors. LABs manage their own sequencing. Whenever a new LAB needs to run, either upon start-up or once the previous LAB completes, each LAB may make a bid for how important it is to run next. A LAB chooses a bid strength depending upon current world conditions, such as time of day, and episodic memory as needed; if the LAB does not care to run, it does not bid at all. An arbitration behavior makes a weighted probability choice among the bids

Narrative Directive Behaviors. By contrast, narrative directive behaviors (NDBs) are more tightly structured collections of sub-behaviors, intended to perform more important and more sophisticated parts of the scenario. NDBs are invoked when an agent is directed to adopt a goal by the Automated Story Director. This is the primary mechanism through which the Automated Story Director prescriptively manipulates the story world to bring about plot progression. For a discussion of the proscriptive direction from the Automated Story Director, see [46].

The dynamic and unpredictable nature of interacting with the participant is handled through a strategy of "mix-in" behaviors, first proposed in [33]. Interaction daemons in each agent await particular inputs from the participant such as dialogue and physical interaction. Special "mix-in" behaviors modify the default logic of the LAB or NDB currently being executed by the agent in order to respond seamlessly and appropriately to the participant.

The generative experience management framework described here has been applied to two prototype systems with vastly different requirements. The first, described in Section 4, is an entertainment system based on the Little Red Riding Hood tale and uses a textual virtual world built on a MOO (Object-oriented Multi User Dungeon). The second, described in Section 5, is a scenario-based military cognitive skills practice environment built on a commercial 3D graphical game engine.

3.3 Authoring

The purpose of a framework is to re-use components in vastly different contexts while minimizing the amount of custom software development. We acknowledge that the virtual world component may need to be re-engineered from application to application depending on the type of immersive experience and art assets required. Our goal is that the Automated Story Director can be re-used by authoring new content: plan operator libraries and exemplar narrative. The set of semi-autonomous character agents will have to change as well, but our use of ABL is designed to allow the specific character-specific behaviors to be authored while reusing the general LAB and NDB goal arbitration process.

One advantage of generative experience management approaches such as [44] [61] and that described in Section 2 over other non-generative experience/drama management approaches is that much of the authoring effort is reduced. Only a single exemplar narrative plan and supporting knowledge about the dynamics of the virtual world need to be authored to apply the Automated Story Director to a new domain context. The system, in an offline, process analyzes the causal dependencies of events in the narrative plan determines all ways in which participant actions can conflict with the exemplar narrative and generates contingency narratives. The result is analogous to a branching story [48]. Pre-scripted branching stories such as those used in computer games and Choose-Your-Own-Adventure books typically have either few decision points or low branching factors (the number of alternatives in any given decision point). One reason for this is the combinatorial complexity of authoring branching stories [10]; as the number of decision points grows, the amount of story content that must be authored grows exponentially. The generative approach used here mitigates this effort by taking the effort narrative adaptation out of the hands of the system designers. The Automated Story Director automatically discovers and implements the "rules" for repairing threatened narratives.

A narrative planner however cannot operate in a vacuum – it requires knowledge about the virtual world in which the narrative is set. The more knowledge the planner has, the greater the number of content variations can be generated. The planner requires a library of plan operators and schemata (referred to as a domain theory). A domain theory describes in abstract terms all actions that are possible for story world characters to perform. It may be the case that the amount of knowledge required by the system is greater than the length of the exemplar narrative or any one plan generated. However, a domain theory is a reusable knowledgebase used by the algorithms described in the article to generate branches recursively. Thus the human author need only supply a single non-branching exemplar narrative and a relatively compact domain theory. For branching narratives that are long and/or have a high branching factor, the amount of narrative content generated from a relatively compact domain theory can quickly exceed the size of the world domain authoring effort that would otherwise need to be authored by hand.

Character agent authoring is knowledge-intensive. The "broad, shallow" approach [6] means that autonomous character behaviors are easy to author, although time-consuming. Knowledge engineers need only focus on the appearance of correct behavior without regard for agent reasoning or deep decision making. The hardest part of authoring "broad, shallow" character agents is providing enough coverage of the space of all behaviors that the agent is never caught in a situation in which it cannot act believably. This coverage is achieved by authoring Local autonomous behaviors (LABs). LABs can be specific to an individual NPC in the case that the character expresses a very unique personality or characteristic. We anticipate that LABs will be general enough be reused or can be customized from generic templates and that over time, large libraries of customizable LABs will exist that provide good coverage of the space of behaviors. For example, a *greeting a customer* behavior could involve the same physical actions from agent to agent, with only dialog variations customized to the agent. Further, some low-level behaviors, such as locomoting, operating a cash register, cleaning the store and so on, are generic and can be re-used from agent to agent, with little or no customization per agent.

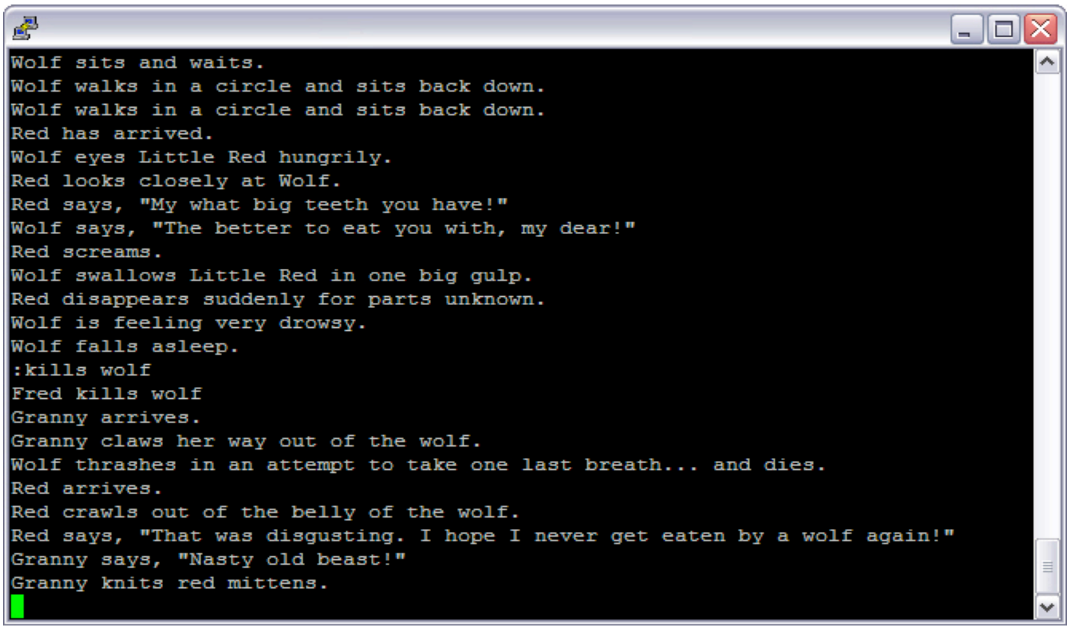
Narrative directive behaviors (NDBs) can be considered a general pre-compiled plan or method for accomplishing world state change. The challenge of authoring NDBs is to consider alternative methods for achieving the same world state condition for a wide variety of possible circumstances. It is possible that narrative directive behaviors can be pre-computed by intelligent agent-based planning systems although we have yet to explore this possibility in any depth. Once authored, we anticipate a wide degree of re-use.

4. Example Entertainment Application: Little Red Riding Hood

Our interactive narrative framework was used to build an interactive narrative system based loosely on the Little Red Riding Hood tale. The virtual world was built on a textual MOO (Object-oriented Multi-User Dimension). Figure 4 shows a screenshot of the Little Red Riding Hood prototype.

The Little Red Riding Hood domain is a difficult domain to convert into an interactive narrative. The principle character, Little Red Riding Hood ("Red" for short) is a victim. The Hunter saves Red, but otherwise does not have much interaction with any other characters. There are not many characters in the story world, meaning there are limited opportunities for the experience manager to direct the participant's experience. To make the Little Red Riding Hood domain more suitable for experience management, we extended the domain to include two extra characters: a fairy, and a monster named Grendel. The fairy has the power to resurrect dead characters. Grendel, like the wolf, is capable of swallowing other characters alive.

The participant plays the role of a hunter – in the screenshot in Figure 4, the participant has chosen the name Fred. As a hunter, the participant has the ability to kill other characters. The choice to make the user play the role of the hunter is non-intuitive. The Little Red Riding Hood character is the "title" character, but she is not the protagonist in the sense that she has very little agency; she is eaten and spends some amount of time in the stomach of the wolf, passively waiting to be rescued. The hunter is the character that rescues both Red and Granny and, as a less developed character, affords the participant the opportunity to exert their own persona. Thus the participant, as hunter, can take a reactive stance and rescue Red and Granny once they are eaten by the wolf, or the participant can take a proactive stance and attempt to exert influence over the virtual world to



```
Wolf sits and waits.
Wolf walks in a circle and sits back down.
Wolf walks in a circle and sits back down.
Red has arrived.
Wolf eyes Little Red hungrily.
Red looks closely at Wolf.
Red says, "My what big teeth you have!"
Wolf says, "The better to eat you with, my dear!"
Red screams.
Wolf swallows Little Red in one big gulp.
Red disappears suddenly for parts unknown.
Wolf is feeling very drowsy.
Wolf falls asleep.
:kills wolf
Fred kills wolf
Granny arrives.
Granny claws her way out of the wolf.
Wolf thrashes in an attempt to take one last breath... and dies.
Red arrives.
Red crawls out of the belly of the wolf.
Red says, "That was disgusting. I hope I never get eaten by a wolf again!"
Granny says, "Nasty old beast!"
Granny knits red mittens.
```

Fig. 4. Screenshot of the Little Red Riding Hood interactive narrative.

keep Red and Granny safe. The latter case is interesting because the Automated Story Director, due to the pre-specified exemplar narrative and islands, attempts to manipulate the participant's experience in the virtual world such that Red and Granny do get eaten and do require rescue from the participant (in accordance with the spirit of the original story).

The exemplar narrative plan is shown in Figure 2. The exemplar narrative is one in which the Wolf meets Red and learns all about Granny. Wolf eats Red and Granny alive, and is then killed by the Hunter (the participant), allowing Red and Granny to escape out of the belly of the Wolf. The outcome is described as:

- Granny has the cake;
- Red is not in the state of being eaten; and
- Granny is not in the state of being eaten.

There are two islands used to enforce a particular narrative structure, should plan adaptation occur. Island 1 represents the fact that Red should become eaten at some point prior to the outcome and Island 2 represents the fact that Granny should become eaten at some point prior to the outcome. Note that the islands do not specify by whom Red and Granny should be eaten. Note also that the narrative plan in Figure 2 describes the plot at a relatively high level of abstraction, so that the details are determined by the semi-autonomous character agents playing the roles of the story world characters. In this case, the semi-autonomous character agents move about the world and improvise scenes with character-specific dialogue when directed. Since the plot points requiring Wolf to eat Red and Granny are unordered relative to each other, the semi-autonomous character agent playing the role of the Wolf is free to decide what order to achieve those goals.

Figure 4 shows a point in the execution of the exemplar narrative where Wolf has already eaten Granny. The Wolf waits for Red to arrive in the cottage and, after some bit of dialogue, eats Red. The dialogue between Red and the Wolf is one of the most recognizable features of the Little Red Riding Hood story. As such, the semi-autonomous character agents are instantiated with behaviors that are capable of performing several different versions of this dialogue (one of which is shown in Figure 4) depending on the conditions of the world and the conditions of the agents. It is important to note that although the dialogue does occur, it is not dictated by Story Director because, from the perspective of the Story Director, dialogue is a detail of character implementation of plot-level events. That is, the Story Director only insists that Red be eaten by the Wolf. However, since the semi-autonomous character agents are actors, they improvise a scene that adds a degree of believability in the process of achieving their prescribed goals.

The next plot point in the exemplar narrative specifies that the Hunter (the participant) is to kill the Wolf. At this point, the narrative is dependent on the participant doing what is expected. After a certain amount of time has passed without the participant killing the Wolf, a failure will trigger and narrative adaptation will occur. However, the participant who is already in the room and who has witnessed the eating of Red, does kill the Wolf.

The principal way in which the participant can create inconsistencies between the state of the virtual world and the narrative plan is by invoking his or her ability as a hunter to kill other characters at times other than that specified in the exemplar narrative plan. For example, the participant can kill the Wolf before it eats Red or Granny. A small portion of the contingency narrative plan tree is shown in Figure 5 (some plan nodes are truncated to show only the plan steps that occur before the islands). The exemplar narrative plan is the root, shown at the left of the figure. Consider the narrative plan node labeled 1 in Figure 5. In this case, the participant has created an inconsistency with the exemplar narrative plan by killing the Wolf before it can eat Red or Granny (and thus before the Automated Story Director can achieve either of the

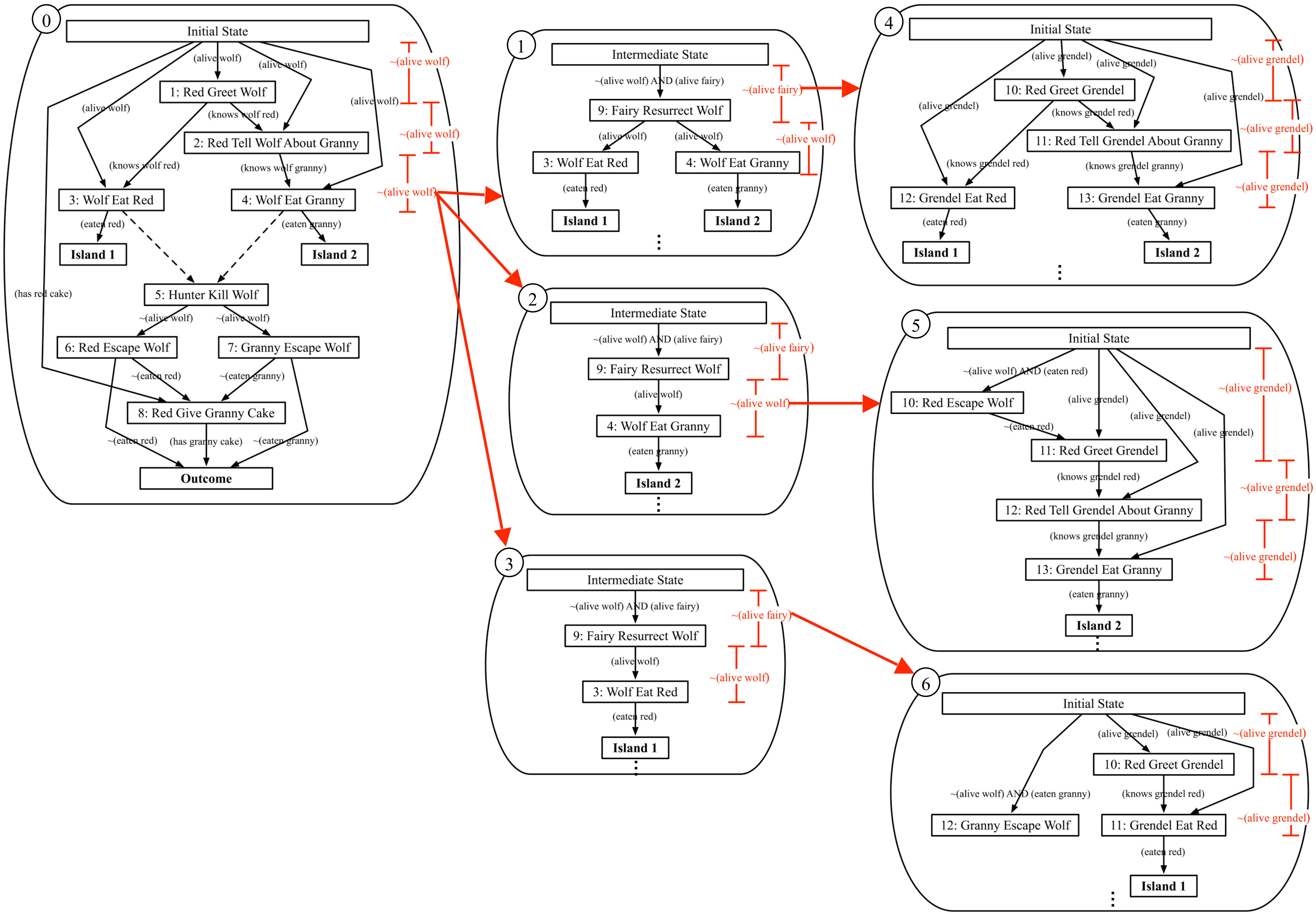


Fig. 5. A portion of the tree of narrative plan contingencies for Little Red Riding Hood.

Fig. 6. A screenshot of the IN-TALE system. The learner (central) is confronting a couple non-player characters.

islands). The simplest narrative adaptation is to have the Fairy resurrect the Wolf. If for some reason the Fairy is also killed by the participant, Grendel can fill the role of the character who eats Red and Granny.

Note that in the exemplar narrative plan, the plot points specifying that the Wolf eat Red and that the Wolf eat Granny are unordered with respect to each other. This creates the possibility of multiple branches based on a race condition between the participant's killing of the Wolf and the Wolf's achievement of the two islands: the Wolf can be killed before eating Red or Granny (node 1); the Wolf can be killed after eating Red but before eating Granny (node 2); or the Wolf can be killed after eating Granny but before eating Red (node 3). If the Wolf is killed after eating Red *and* Granny, the Automated Story Director recognizes this as a fulfillment of the participant action, Hunter kills Wolf. Each possible ordering of events in the race condition results in a slightly different narrative adaptation.

5. Example Educational Application: IN-TALE

As stated earlier, drama management is a term coined to describe an approach to interactive entertainment. Experience management is a generalization of drama management that does not strictly require dramatic narrative. Interactive narrative has previously been applied to educational and training contexts (c.f. [4] [29] [30]). In this section, we describe a prototype built on our generative experience management framework for education and training. Experience management holds potential as an approach to interactive narrative for education and training. As discussed in Section 1.3, narrative is often used in education and training. Narrative plays a role in the access and transfer tacit expert knowledge [7] [52], case-method teaching practices [23], and can facilitate learning by doing [1].

The Interactive Narrative Tacit Adaptive Leader Experience (IN-TALE) prototype system [45] [46] is an example of how interactive narrative using a generative experience manager can be applied to a training curriculum for "soft skills" – a term the U.S. military uses to refer to skills such as leadership, situation awareness, cultural awareness, decision-making, and other cognitive skills. Research on leadership development shows that expertise is gained through experience and by taking the time to reflect on the lessons learned from an episode [52].

The IN-TALE interactive narrative provides a practice environment to accompany a more conventional instructional curriculum. The learner, playing a leadership role in a simulation of a military exercise, is tasked with applying skills and problem solving to achieve a particular effect. For example, the learner might be told that he or she is in charge of maintaining law and order in a marketplace during a foreign deployment. Figure 6 shows a screenshot of the In-TALE system.

As an interactive narrative application for education and training, exemplar narratives must support the pedagogical objectives of an instructor and/or curriculum. Instead of structuring the exemplar narrative around dramatic principles such as those that might be used in an entertainment-based system, we propose that the exemplar narrative be structured around *relevant learning situations* (RLS). A relevant learning situation is a world state that presents the learner with a dilemma or an opportunity to perform a skill or make an observation. Computationally, relevant learning situations manifest themselves in the exemplar narrative as *islands* – intermediate goal states that must be established by events in the plan somewhere between the beginning and the end. To support relevant learning situations, the Automated Story Director is modified in the following ways. Relevant learning situations are provided as input in addition to the exemplar narrative (relevant learning situations are also part of the exemplar narrative sequence).

5.1 IN-TALE Exemplar Narrative

The following describes an exemplar narrative that might be provided to IN-TALE to provide a practice environment for situational and cultural awareness. The learner plays the role of a Captain in charge of maintaining law and order in a marketplace during a foreign deployment.

The scenario is expected to unfold as follows. While the learner is engaged in daily procedures concerning the operations of the marketplace, the learner has opportunities to

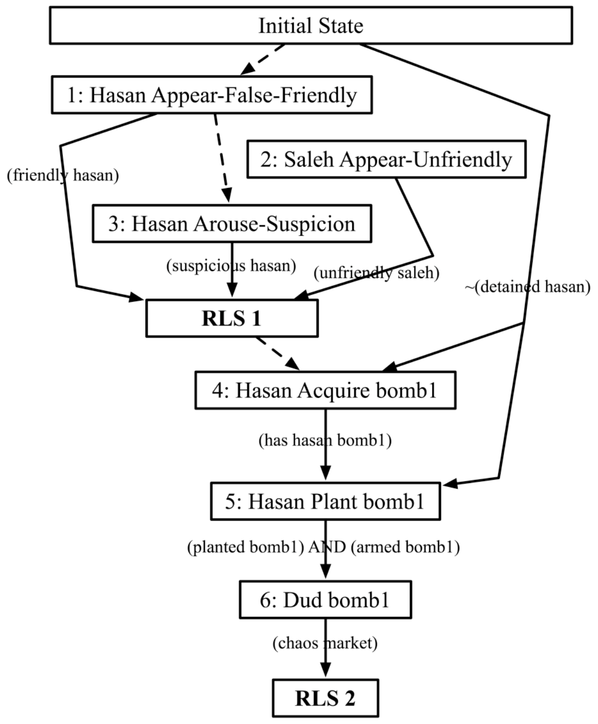


Fig. 7. Example narrative plan for the IN-TALE system.

interact with various merchants who regularly use the marketplace. One merchant, Hasan, appears to be friendly and pro-American (although if carefully observed, he performs subtle behaviors meant to arouse the learner's suspicions). Another merchant, Saleh, appears to be unfriendly and to hold anti-American sentiment. After the learner has had opportunities to become familiar with the other story world characters, Hasan – who despite outwardly favorable behavior and appearance has an ulterior motive – sneaks away from the marketplace and later returns concealing an explosive device that he obtained from a co-conspirator named Ayad. The device is planted in the marketplace. Shortly afterwards, the bomb goes off. When it goes off, it is a dud, creating a lot of smoke and noise but no casualties. The marketplace is none-the-less plunged into a state of social chaos, disorder, and panic, resulting in economic harm. The scenario continues from this point, but is outside the scope of this paper.

There are two relevant learning situations present in the example exemplar narrative. The first occurs early on in the narrative sequence when the characters of Hasan and Saleh are established. This relevant learning situation is an opportunity to observe that Saleh, despite outward appearances, is harmless, and that Hasan, despite outward appearances, is a complex individual who might harbor ill intentions towards the learner. The second relevant learning situation occurs at the end of the example and describes a world state in which the marketplace is in a state of chaos. The first relevant learning situation focuses on situational awareness. The second relevant learning situation requires the learner to practice restoring order and handling consequences in the context of a continuing scenario. Figure 7 shows the plan representation for this exemplar narrative.

Note that if the learner performs exceptionally in the first relevant learning situation, it is possible for the learner to deduce that Hasan is planning an attack. If this happens, due to the open-ended interactivity afforded by the system, it is also possible that the learner can intercept Hasan and detain him before the explosive device can be armed and planted. Should this happen, the second relevant learning situation – chaos in the marketplace – can be averted. While this is potentially a realistic outcome of the learner's actions, the learner's actions inadvertently make it impossible, or at least implausible, to establish the second relevant learning situation, thus reducing the pedagogical effectiveness of the system and averting part of the instructional intent of the system. The Automated Story Director is able to detect this predicament and invoke the narrative generation process to adapt the narrative sequence so that the second relevant learning situation is achieved in a different way – possibly by using the detention of Hasan to provoke a riot as an alternative way of establishing a state of chaos in the marketplace.

5.2 IN-TALE Example

Continuing the example, suppose IN-TALE is instantiated with the exemplar narrative plan shown in Figure 7. The islands are listed as RLS1 and RLS2 (i.e., relevant learning situation 1 and relevant learning situation 2), where the first relevant learning situation represents that an opportunity to make an observation about the non-player characters has been achieved, and the second relevant learning situation is that the marketplace is in chaos and must be handled. Figure 8 shows a portion of the tree of contingency narrative plans generated automatically from the exemplar narrative. Note that the actual contingency tree generated by the Automated Story Director can contain in excess of 2000 nodes, only a few of which are actually shown in Figure 8.

The example considers two ways in which the learner can create an inconsistency between the virtual world and the exemplar narrative plan:

- Hasan is detained and thus unable to perform further actions in the world; and
- bomb1 is found and disarmed before it goes off.

There are three intervals in the exemplar narrative plan in which these inconsistencies are possible. The first occurs if Hasan is detained before he can acquire the bomb. The narrative plan that repairs this inconsistency is not shown, but the new narrative plan has the local Magistrate release Hasan from detention, ostensibly because he has not perpetrated any crime. Second, a potential inconsistency can occur if Hasan is detained after acquiring the bomb, but before he can plant it. This contingency narrative plan (node 1) also has the Magistrate release Hasan. However, if Hasan is searched by the learner (before or after he is detained) and is found to be concealing a bomb, he can be designated a criminal. If designated a criminal, Hasan cannot be released from detention, causing a cascading branch to node 3. In this case, the Automated Story Director has found a narrative that obtains the second relevant learning situation – chaos in the marketplace – in a radically different way. Specifically, Hasan's co-conspirator, Ayad, uses Hasan's detention as an excuse to cause a riot.

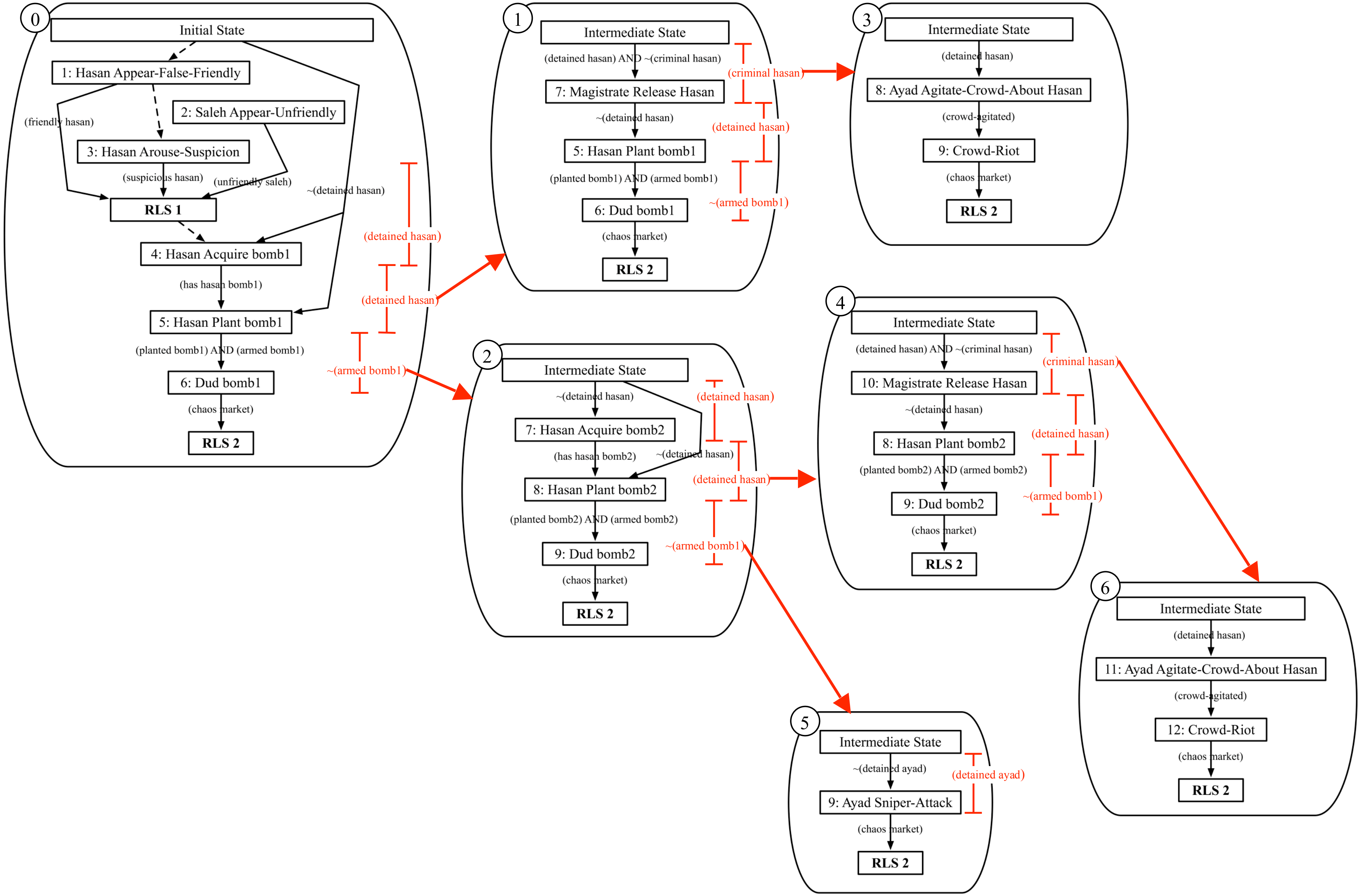


Fig. 8. A portion of the tree of narrative plan contingencies for IN-TALE.

The third interval in which the learner can force the Automated Story Director to adapt the narrative can occur if he or she finds the planted bomb and calls in a bomb-squad to disarm it. In this case, the simplest adaptation to the narrative is to have Hasan acquire a second bomb and plant it (node 2). Note that many of the same patterns of potential inconsistencies (e.g detaining characters and disarming bombs) seen in the exemplar narrative appear to re-occur in node 2. However, one distinction is that there is not an infinite supply of bombs in the story world, so the Automated Story Director eventually turns to alternative strategies for bringing about the second relevant learning situation (RLS2). For example, node 5 involves a sniper attack on the marketplace.

There is no reason why the Automated Story Director need repeat patterns of narrative adaptations. In the current implementation, the Automated Story Director looks for the simplest adaptations. The assumption underlying this is that the simplest adaptations are those that will be most similar to the exemplar narrative and that the exemplar is the "best" experience (from the perspective of the instructor and/or curriculum) the learner can have. Currently we are investigating whether the automated narrative generation process used by the Automated Story Director can incorporate more sophisticated pedagogical reasoning in order to provide richer narrative experiences.

6. Related Work

We describe interactive narrative as an approach to interactive entertainment in which a system attempts to tell a story to an interactive participant such that the participant is afforded opportunities to make decisions or perform actions that directly affect the direction and/or outcome of the story. There are numerous examples of systems that qualify as interactive narrative systems. The work presented here has the most in common with other systems that use artificial intelligence to implement interactive narratives. Space precludes a comprehensive comparison of the generative experience management framework described here to all other interactive narrative systems. In the remainder of this section we discuss those systems with interesting similarities and differences.

The Carnegie Mellon University Oz Project [6] [25] [59] and follow on work (c.f, [36] [37] [49] [50]) use a drama management approach based on plot graphs. A plot graph is a partial ordering of all possible plot points, some of which will occur and some of which will not. The drama manager searched for and executed sequences of plot points and interventions (events that would help or hinder the participant). The Automated Story Director makes use of partially-ordered plans that must be completely executed; if a plan cannot be executed, then a new plan is generated. Theoretically, a plot graph (plus interventions) and the Automated Story Director's tree of contingency plans can both be represented as branching narratives (see [48] for a sketch of a proof of the latter). The distinction is that a plot graph is hand authored while the tree of contingency plans is automatically generated.

The *Façade* interactive drama [32] [33] implements an approach to drama management based on beats – small segments of story, typically involving actions/reactions for two characters. The Little Red Riding Hood and IN-TALE prototypes are relatively simple compared to the full complexity (in terms of branches) of a complete system such as *Façade* [32] [33]. However, these prototypes illustrate the capabilities and advantages of using a generative experience management approach. Specifically, the Automated Story Director (a) automatically discovers and implements the "rules" for repairing threatened narratives, and (b) automatically generates all possible plots that the participant's experience can take the form of. This greatly reduces the amount of human authoring effort necessary to deploy an interactive narrative for entertainment or education. Like *Façade*, dialogue and specifically performed behaviors (e.g. gestures) are manually authored as re-usable and re-assemble-able chunks that are selected by character agents in real-time.

Many of the techniques used in the Automated Story Director framework were pioneered for the Mimesis system [44] [61], an architecture for AI-based interactive narratives. Both systems use a planner to project a narrative into the future and re-plan as necessary. The Mimesis system generates the first narrative, whereas the Automated Story Director relies on a hand-authored exemplar and generates only the branches. Further differences exist between systems with regard to how re-planning works, the Automated Story Director using a tiered re-planning scheme. The most notable difference is how virtual characters are controlled. In the Mimesis system, virtual characters directly execute operators in the plan whereas the Automated Story Director framework translates plan operators into goals to be disseminated to semi-autonomous agents. The significance of this is that character agents in the Automated Story Director framework are capable of improvising activity when not being directed in order to appear more animate and life-like.

The FearNot! system [4] is a strong autonomy system used to teach students about bullying. Autonomous agents play the roles of characters and situations relevant to bullying emerge from the interactions between characters. FearNot! uses an agent called a *story facilitator* [15] that sets up scenes and characters for narrative emergence to occur. In many ways the FearNot! story facilitator and the Automated Story Director provide similar services. However, the story facilitator works with authored "scenes" that are then elaborated upon by autonomous agents. The Automated Story Director generates sequences of high-level events, several of which may be perceived to make up a "scene" that are then elaborated upon by semi-autonomous character agents.

In general, the distinguishing trait between the work described here and most other drama management systems is the amount of narrative content that must be hand-authored *a priori*. By generating narrative content in response to user actions, generative experience managers hold the potential to deliver more interactive and more varied experiences. It should be noted that there are other *generative* interactive narrative systems. For example, [5] describes an interactive narrative that uses planning – in much the same way as the

Automated Story Director and Mimesis systems use planning – to create dilemma situations in an open-ended narrative. It should also be noted that some emergent narrative systems such as [14] are considered generative. Detailed consideration of emergent narrative systems is beyond the scope of this paper.

7. Limitations and Future Work

Our research has revealed to us several limitations to be addressed through future work. In this section, we discuss some of the limitations of our framework and how we intend to address them through future work.

Narrative Generation. The most significant limitation is that the success of the generative experience management framework is linked to the technical capability to automatically generate narratives. However, it is our observation is that current state-of-the-art in narrative generation is not yet sufficient. In particular, general problem solvers such as planners tend to search for efficient solutions to a problem. However, it is often the case that the most efficient solution to a problem is not the best *story*. Specialized narrative planners such as [42] [47] may produce better results. Case-based approaches to narrative generation (c.f., [16] [39] [55]) may allow a narrative generator to favorably utilize knowledge from previous narratives.

Heuristic functions are used to evaluate whether a narrative plan is good. Additional heuristics must be used to determine whether an adaptation is good. Currently there is no sufficient computational model of a good narrative. Due to the limitations of narrative planning and the lack of heuristics the Automated Story Director tends to favor the simplest adaptations and different branches tend to be repetitive, which a few significantly different and interesting adaptations appearing when no easier adaptation is possible. Future work involves building off of narrative generation research described above and addressing the issue of heuristics.

Intelligent Tutoring. We believe that the incorporation of an intelligent tutoring system (ITS) into interactive narratives for education and training will benefit the learner by supporting the learner with individualized instruction. An intelligent tutoring system is an intelligent agent that, like a human tutor, engages with a student to scaffold learning [51] [58]. The IN-TALE prototype described in Section 5 is an example of an interactive narrative with pedagogical purposes. IN-TALE is not an ITS; it is a practice environment meant to provide a degree of interactivity to support the practice of ill-defined cognitive skills such as situational awareness, cultural awareness, leadership, and decision-making. IN-TALE can manipulate a participant into relevant learning situations, but it cannot make the participant learn. Consequently, IN-TALE does not – nor is meant to – teach. We believe that intelligent tutoring and generative experience management are complimentary technologies. See [43] for an informal analysis of the compatibilities of integrating intelligent tutoring and interactive narrative.

User Modeling. We have also identified user modeling as an important consideration for interactive narrative systems. Without a model of the user, the Automated Story Director over-generates, meaning it generates all possible narrative sequences that can occur, no matter how improbable. We use a very simple, static, handcrafted user model to prune the contingency tree of branches that are unlikely to be taken. However, more robust and intelligent approaches will be advantageous. The approach described in [28] uses a player model to predict whether the user will perform actions that transgress from the space of acceptable narrative experiences. This approach to user modeling is not appropriate for a generative experience management system because the space of acceptable narrative experiences is modified dynamically. Learning and case-based approaches to user modeling in interactive narratives have also been explored (c.f., [37] and [50]). We intend to explore user modeling as part of our framework in the future. As it pertains to the IN-TALE prototype, user modeling is often used to track what learners know so a system can make more intelligent decisions about how to scaffold and individualize learning.

8. Summary and Conclusions

We present a framework for developing interactive narratives that balance the desired properties of the participant's narrative experience against the participant's perception of self-agency. The framework involves a generative experience manager called the Automated Story Director that attempts to manipulate the virtual world to conform to an exemplar narrative sequence that encodes particular provided properties. These properties can be dramatic or pedagogical. The framework is illustrated with two example interactive narrative systems. The first is an interactive version of the Little Red Riding Hood story, meant for entertainment purposes. The second, the Interactive Narrative Tacit Adaptive Leader Experience (IN-TALE), demonstrates the experience management framework applied to a practice environment for educating cognitive skills such as situational awareness, cultural awareness, leadership, and decision-making under pressure.

The generative experience management framework is currently at prototype stage. The framework was initially developed for the IN-TALE system. The IN-TALE exemplar scenario was developed to test the algorithms and has not been vetted for pedagogical significance, nor have any of the generated alternative narrative branches. As noted in Section 7, the generation algorithms tend to favor the simplest adaptations and also tend to be repetitive. In some ways this is favorable because the closer an adaptation is to the original exemplar, the more likely that that adaptation will adhere to the original authorial intent. However, the simplest adaptation often results in repetitive storylines. For example, if bomb-planting is disrupted by the user, the Automated Story Director will often try a different way to plant the bomb. Users sometimes perceive this as absurd, obsessive, or frustrating.

The IN-TALE prototype generates over 2000 nodes in the contingency tree (we cap the depth of the tree at 6 levels for testing purposes, meaning that the user can only force an adaptation of the storyline 5 times in one session). Each node represents a complete alternative storyline, although most nodes represent minor variations on each other where only one or two details differ. The average branching factor of the contingency tree is 2.87, meaning that during execution, the user has on average nearly three ways in which he or she can act to interfere with the currently executing narrative plan. Because of the continuous real-time execution, these acts can be performed at any time. Of course the IN-TALE scenario is a test case and actual scenarios are expected to be much more complex, in terms of narrative plan length and branching factor.

In the Fall of 2006, the International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) held an authoring workshop in which participants were challenged to modify their interactive systems to the Little Red Riding Hood domain. The framework, initially developed for the IN-TALE training domain was converted to the new domain in three weeks, including a new exemplar narrative, new domain theory knowledge from which narrative branches could be generated, and new character agents. A programmer with AI planner experience authored a domain theory sufficient to cover the range of possible events in the fairy tale world in a couple of days. The majority of the time was spent creating new character agents since there was little that could be reused from the IN-TALE domain. The system was demoed live at the workshop. Development was simplified by the use of a textual virtual world, allowing us to bypass the challenges of graphical character animations.

Anecdotal evidence indicates that most players of the entertainment-based Little Red Riding Hood system are content with watching the narrative unfold, and only take action when it comes time to rescue Little Red and Granny. Other players come to treat it as a game in which they try to stop the Wolf (and Grendel) from eating Little Red and Granny in the first place, even though the narrative arc leads to their rescue. That is, players learn to take actions that force the Automated Story Director to adapt until the Automated Story Director is forced to abandon its goals of having Little Red and Granny eaten (see Section 2.3, strategy iii). Replayability becomes an important issue when this happens.

Immersive virtual worlds, whether textual or graphical, have the potential to tap into the human propensity to organize our experiences as narrative. This is especially true when the virtual world affords social interaction – even when the social interactors are computer-controlled, embodied agents. Guidance, however, is sometimes required to increase the likelihood that an experience is cognitively reconstructed as narrative. Drama managers – and their more general form, experience managers – offer the ability to guide a participants experience in a virtual world without reducing the participants ability to act freely to pursue his or her goals, intentions, and desires. A *generative* approach to experience management is especially useful because a generative experience manager can engage with a participant or learner to accommodate his or her actions without creating undue burden on system authors and designers who would otherwise have to craft large amounts of narrative content to cover all contingencies.

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. Special thanks to Chirag Merchant and Carlos León for their development work. Further thanks are extended to the anonymous reviewers, whose constructive feedback helped to improve this article.

References

[1] C Andersen. Learning in “As-If” Worlds: Cognition in Drama in Education. *Theory into Practice*, Vol. 43, No. 4, 2004.

[2] R Aylett. Narrative in virtual environments – Towards emergent narrative. In M. Mateas and P. Sengers (Eds.) *Narrative Intelligence: Papers from the AAAI Fall Symposium* (Technical Report FS-99-01). AAAI Press, Menlo Park, 1999.

[3] R Aylett. Emergent Narrative, social immersion and “storification.” *Proc. of the 1st Int. Workshop on Narrative and Interactive Learning Environments*, 2000.

[4] R Aylett, S Louchart, J Dias, and A Paiva. FearNot! an Experiment in Emergent Narrative. *Proc. of the 5th Int. Conf. on Intelligent Virtual Agents*, 2005.

[5] H Barber and D Kudenko. Dynamic Generation of Dilemma-Based Interactive Narratives. *Proc. of the 3rd Conf. on Artificial Intelligence and Interactive Digital Entertainment*, 2007.

[6] J Bates. Virtual Reality, Art, and Entertainment. *Presence: The Journal of Tele-operators and Virtual Environments*, Vol. 1, No. 1, 1992.

[7] R F Baumeister and L S Newman. How Stories Make Sense of Personal Experiences: Motives that Shape Autobiographical Narratives. *Personality and Social Psychology Bulletin*, Vol. 20, No. 6,1994.

[8] D Blair and T Meyer. Tools for an Interactive Virtual Cinema. In R Trappl and Paolo Petta (Eds.) *Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents*. Springer, 1997.

[9] B Blumberg, T Galyean. Multi-Level Control for Animated Autonomous Agents: Do the Right Thing… Oh, not that… In R Trappl and P Petta (Eds.) *Creating Personalities for Synthetic Actors*. Springer, 1997.

[10] A Bruckman. *The Combinatorics of Storytelling: Mystery Train Interactive*. Unpublished manuscript, http://www.cc.gatech.edu/~asb/papers/combinatorics-bruckman-90.pdf (retrieved 1 Nov. 2007), 1990

[11] J Bruner. *Acts of Meaning*. Harvard University Press, Cambridge, 1990.

[12] J Bruner. The Narrative Construction of Reality. *Critical Inquiry*, Vol. 18, No. 1, 1991, p. 1-21.

[13] J Cassell. Embodied Conversational Agents: Representation and Intelligence in User Interface. *AI Magazine*, Vol. 22, No. 3, 2001.

[14] M Cavazza, F Charles, and S Mead. Planning Characters' Behaviour in Interactive Storytelling. *Journal of*

Visualization and Computer Animation, Vol. 13, 2002.

[15] R Figueiredo, J Dias, A Paiva, R Aylett, S Louchart. Shaping Emergent Narratives for a Pedagogical Application. *Proceedings of the 4th International Conference on Narrative and Interactive Learning Environments*, 2006.

[16] P Gervás, B Díaz-Agudo, F Peinado, and R Hervás. Story Plot Generation Based on CBR. *Journal of Knowledge-Based Systems*, vol. 18, No. 4-5, 2005.

[17] R Gerrig. *Experiencing Narrative Worlds: On the Psychological Activities of Reading*. Yale University Press, New Haven, 1993.

[18] R Gerrig, Narrative Though? *Personality and Social Psychology Bulletin*, Vol. 20, No. 6, 1994.

[19] R Gerrig and D Rapp. Psychological processes underlying literary impact. *Poetics Today*, Vol. 25, No. 2, 2004.

[20] M C Green and T Brock. The Role of Transportation in the Persuasiveness of Public Narratives. *Journal of Personality and Social Psychology*, Vol. 79, No. 5, 2000.

[21] J Harris and R M Young. Proactive Mediation in Plan-Based Narrative Environments. *Proc. of the 5th Int. Working Conference on Intelligent Virtual Agents*, 2005.

[22] B Hayes-Roth and F Hayes-Roth. A Cognitive model of planning. *Cognitive Science*, Vol. 3, 1979.

[23] R W Hill, A S Gordon, and J M Kim. Learning the Lessons of Leadership Experience: Tools for Interactive Case Method Analysis. *Proc. of the 24th Army Science Conference*, 2004.

[24] H Jenkins. Game Design as Narrative Architecture. In N Wardrip-Fruin and P Harrigan (Eds.) *First Person: New Media as Story, Performance*, Game. MIT Press, Cambridge, 2004.

[25] M Kelso, P Weyhrauch, J Bates. Dramatic Presence. Presence: *The Journal of Teleoperators and Virtual Environments*, Vol. 2, 1993.

[26] C Knoblock. Generating Parallel Execution Plans with a Partial-Order Planner. *Proc. of the 2nd Int. Conference on Artificial Intelligence Planning Systems*, 1994.

[27] A B Loyall. *Believable Agents: Building Interactive Personalities*. Ph.D. Dissertation, Carnegie Mellon University, 1997.

[28] B Magerko. Evaluating Preemptive Story Direction in the Interactive Drama Architecture. *Journal of Game Development*, vol. 2, no. 3, 2005.

[29] B Magerko, R Wray, L Holt, and B Stensrud. Customizing Interactive Training through Individualized Content and Increased Engagement. *Proc. of the 2005 Interservice/Industry Training, Simulation, and Education Conference*, 2005.

[30] S Marsella, W L Johnson, and K LaBore. Interactive Pedagogical Drama. *Proc. of the 4th International Conference on Autonomous Agents*, 2000.

[31] M Mateas and P Sengers. Narrative Intelligence. In M Mateas and P Sengers (Eds.) *Narrative Intelligence: Papers from the AAAI Fall Symposium* (Technical Report FS-99-01). AAAI Press, Menlo Park, 1999.

[32] M Mateas and A Stern. Towards Integrating Plot and Character for Interactive Drama. Social Intelligent Agents: *The Human in the Loop: Papers from the AAAI Fall Symposium* (Technical Report FS-00-04), 2000.

[33] M Mateas and A Stern. *Architecture, Authorial Idioms and Early Observations of the Interactive Drama Façade* (Technical Report CMU-CS-02-198). School of Computer Science, Carnegie Mellon University, 2002.

[34] M Mateas and A Stern. A Behavior Language: Joint Action and Behavior Idioms. In H Prendinger and M Ishizuka (Eds.) *Life-like Characters: Tools, Affective Functions and Applications*. Springer, 2004.

[35] Y Miao, U Hoppe, and N Pinkwart. Situation Creator: A Pedagogical Agent Creating Learning Opportunities. *Proc. of the 13th Int. Conf. on Artificial Intelligence in Education*, 2007.

[36] M Nelson, and M Mateas. Search-Based Drama Management in the Interactive Fiction Anchorhead. *Proc. of the 1st Conf. on Artificial Intelligence and Interactive Digital Entertainment*, 2005.

[37] M Nelson, M Mateas, D L Roberts, and C Isbell. Declarative Optimization-Based Drama Management in Interactive Fiction. *IEEE Computer Graphics and Applications*, Vol. 26, No. 3, 2006.

[38] E Pednault. *Towards a Mathematical Theory of Plan Synthesis*. Ph.D. Dissertation, Stanford University, 1986.

[39] R Pérez y Pérez and M Sharples. MEXICA: A Computer Model of a Cognitive Account of Creative Writing. *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 13, 2001.

[40] G Prince. *A Dictionary of Narratology*. University of Nebraska Press, Lincoln, 1989.

[41] J Rickel and W L Johnson. Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control. *Applied Artificial Intelligence*, Vol. 13, 1999.

[42] M O Riedl. *Narrative Generation: Balancing Plot and Character*. Ph.D. Dissertation, North Carolina State University, 2004.

[43] M O Riedl, H C Lane, R Hill, and W Swartout. Automated Story Direction and Intelligent Tutoring: Towards a Unifying Architecture. *Proc. of the 2005 AIED Workshop on Narrative Learning Environments*, 2005.

[44] M O Riedl, C J Saretto, and R M Young. Managing Interaction between Users and Agents in a Multi-Agent Storytelling Environment. *Proc. of the 2nd Int. Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003.

[45] M O Riedl and A Stern. Believable Agents and Intelligent Story Adaptation for Interactive Storytelling. *Pro. of the 3rd Int. Conf. on Technologies for Interactive Digital Storytelling and Entertainment*, 2006.

[46] M O Riedl and A Stern. Failing Believably: Toward Drama Management with Autonomous Actors in Interactive Narratives. *Proc. of the 3rd Int. Conf. on Technologies for Interactive Digital Storytelling and Entertainment*, 2006.

[47] M O Riedl and R M Young. An Intent-Driven Planner for Multi-Agent Story Generation. *Proc. of the 3rd Int. Joint Conference on Autonomous Agents and Multi Agent Systems*, 2004.

[48] M O Riedl and R M Young. From Linear Story Generation to Branching Story Graphs. *IEEE Computer Graphics and Applications*, Vol. 26, No. 3, 2006.

[49] D L Roberts, C Strong, and C Isbell. Using Feature Value Distributions to Estimate Player Satisfaction Through an Author's Eyes. *Proc. of the 2007 AAAI Fall Symposium on Intelligent Narrative Technologies*, 2007.

[50] M Sharma, S Ontañón, C Strong, M Mehta, and A Ram. Towards Player Preference Modeling for Drama Management in Interactive Stories. *Proc. of the 20th International Florida Artificial Intelligence Research Society Conference*, 2007.

[51] V J Shute and J Psotka. Intelligent Tutoring Systems: Past, Present and Future. In D Jonassen (Ed.) *Handbook of Research on Educational Communications and Technology*. Scholastic Publication, 1996.

[52] R Sternberg, G Forsythe, J Hedlund, J Horvath, R Wagner, W Williams, S Snook, and E Grigorenko. *Practical Intelligence in Everyday Life*. Cambridge University Press, New York, 2000.

[53] W Swartout, R Hill, J Gratch, W L Johnson, C Kyriakakis, C LaBore, R Lindheim, S Marsella, D Miraglia, B Moore, J Morie, J Rickel, M Thiebaux, L Tuch, R Whitney, and J Douglas. Toward the Holodeck: Integrating Graphics, Sound, Character and Story. *Proc. of the 5th Int. Conf. on Autonomous Agents*, 2001.

[54] N Szilas. IDtension: A Narrative Engine for Interactive Drama. *Proc. of the 2nd Int. Conf. on Technologies for Interactive Digital Storytelling and Entertainment*, 2003.

[55] S Turner. *The Creative Process: A Computer Model of Storytelling*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1994.

[56] K VanLehn. The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, vol. 16, 2006.

[57] D Weld. An Introduction to Least Commitment Planning. *AI Magazine*, Vol. 15, 1994.

[58] E Wenger. *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann, San Francisco, 1987.

[59] P Weyhrauch. *Guiding Interactive Fiction*. Ph.D. Dissertation, Carnegie Mellon University, 1997.

[60] R M Young, M E Pollack, and J D Moore. Decomposition and Causality in Partial-Order Planning. *Proc. of the 2nd Int. Conf. on Artificial Intelligence and Planning Systems*, 1994.

[61] R M Young, M O Riedl, M Branly, A Jhala, R J Martin, and C J Saretto. An Architecture for Integrating Plan-Based Behavior Generation with Interactive Game Environments. *Journal of Game Development*, Vol. 1, 2004.

Author Bios

Mark O. Riedl is a research scientist at the University of Southern California Institute for Creative Technologies (ICT). His research interests include AI, narrative and story generation, interactive narrative, virtual cinematography, intelligent virtual agents, and intelligent user interfaces. Mark received his Ph.D. from North Carolina State University in 2004 where he studied a variety of human-computer interaction technologies, from computational modeling of human social navigation behavior to the use of storytelling to guide intelligent agent behavior. His dissertation research investigated AI techniques for generating stories with recognizable plot structures and character fidelity. At the ICT, Mark is investigating interactive narrative techniques for education and training. He has been involved in research to make AI computer game opponents more intelligent and adaptable, to generate and adapt stories to guide trainee experiences in virtual environments, to generate computer graphics based short movies, and to create intelligent authoring tools for novice storywriters.

Andrew Stern is a designer, writer and engineer of personality-rich, AI-based interactive characters and stories. In 2005 he completed the interactive drama Façade, a 5-year art/research collaboration with Michael Mateas. Façade won the Grand Jury Prize at the 2006 Slamdance Independent Game Festival. Andrew recently co-founded the game studio Procedural Arts to develop next-generation interactive story games. Previously Andrew was a lead designer and software engineer at PF.Magic, developing Virtual Babyz, Dogz, and Catz, which sold over 3 million units worldwide. Press for these projects includes The New York Times, The Atlantic Monthly, The Economist, Newsweek, Wired and AI Magazine. Andrew has presented and exhibited work at the Game Developers Conference, Independent Games Festival, SIGGRAPH, ISEA, Digital Arts and Culture, DiGRA, TIDSE, AAAI symposia, Autonomous Agents, Intelligent User Interfaces, and several art shows internationally. He blogs at grandtextauto.org.

Don M. Dini is the Owner and Chief Technical Officer of Method in Mind, a Los Angeles, CA based company focused on the application of artificial intelligence to business and consumer software. Prior to this he was a Research Programmer at USC's Institute for Creative Technologies. There, he worked on many problems relating to artificial intelligence for virtual environments, but focusing on planning and execution systems, analogical reasoning, and the simulation of large scale urban populations. He holds an MS in Computer Science from the University of Southern California and a BS in Physics from the University of Illinois at Urbana-Champaign.

Jason M. Alderman contributed to Automated Story Director research at the University of Southern California Institute for Creative Technologies (ICT) during two consecutive summer internships. In addition to coding user interface elements and writing behaviors, he drew upon six years of military experience to help develop scenarios for the teachable narratives. Jason has a bachelor's degree in Physics from the US Air Force Academy, and he received an MS in Information Design and Technology from Georgia Tech in 2006. Since completing his Masters, Jason designed educational games for preschool children for the Waterford Research Institute, and currently works as an interaction designer and consultant on healthcare software with Veracity Solutions.