Believable Agents and Intelligent Story Adaptation for Interactive Storytelling

Mark O. Riedl[1] and Andrew Stern[2]

[1] University of Southern California, Institute for Creative Technologies, 13274 Fiji Way,
Marina del Rey, California, USA
[2] Procedural Arts LLC, Portland, Oregon, USA
riedl@ict.usc.edu, andrew@proceduralarts.com

Abstract. Interactive Narrative is an approach to interactive entertainment that enables the player to make decisions that directly affect the direction and/or outcome of the narrative experience being delivered by the computer system. Interactive narrative requires two seemingly conflicting requirements: coherent narrative and user agency. We present an interactive narrative system that uses a combination of narrative control and autonomous believable character agents to augment a story world simulation in which the user has a high degree of agency with narrative plot control. A drama manager called the Automated Story Director gives plot-based guidance to believable agents. The believable agents are endowed with the autonomy necessary to carry out directives in the most believable fashion possible. Agents also handle interaction with the user. When the user performs actions that change the world in such a way that the Automated Story Director can no longer drive the intended narrative forward, it is able to adapt the plot to incorporate the user's changes and still achieve dramatic goals.

1 Introduction

Simulation has been, and continues to be, an important part of computer ente-Srtainment. Computer games such as *The Sims* and *SimCity* are simulations in the strictest sense. The initial parameters, a model of state change, and legal player moves are the forces that drive the user's experience. Real-time strategy games are simulations of combat and have been used effectively for entertainment and for military training. First- and third-person action/adventure games also rely on simulation to a lesser extent. Each mission can be considered a simulation of physics, weapon effects, opponent movements, etc. One difference between action/adventure games and other games such as *The Sims* or *SimCity* is the use of story to constrain the player's experience to a particular narrative path.

It is a commonly held belief that narrative and interactivity are diametrically opposed, meaning one can have story or one can have interactivity but not both simultaneously. The most common role of story in computer games is to provide "glue" between missions. Modern computer games consist of interleaved periods of interactive play and cut scenes – short non-interactive scenes that transition from one mission to the next, providing the player with goals and motivation for the next

S. Göbel, R. Malkewitz, and I. Iurgel (Eds.): TIDSE 2006, LNCS 4326, pp. 1 – 12, 2006.

segment of game play. In this mode of alternating between game play and cut scenes, story elements and simulation are kept strictly separate. This is one manifestation of the trade-off between control and coherence [10; 14]. On one hand, the player wants control to make decisions for the player's character. On the other hand, game designers want the player to experience a coherent narrative progression.

Interactive Narrative is an approach to interactive entertainment that enables the player to make decisions that directly affect the direction and/or outcome of the narrative experience being delivered by the computer system. We are building an Interactive Narrative System for training and entertainment that mixes story and simulation. But why use simulation? After all, it has been demonstrated that Interactive Narratives can be constructed out of branching story sequences such as those used by the Choose-Your-Own-Adventure books. The reason is that simulation provides a realistic, continuous, life-like modality for interaction that can result in a more immersive and compelling experience for players than simple branching stories. The realism of simulation is also beneficial to educational games that provide a realistic learning experience, in addition to a coherent narrative progression. Mateas and Stern [9] lay out a continuous spectrum of technical approaches to interactive narrative ranging from *strong autonomy* to *strong story*. The strong autonomy approach advocates that interactive narratives be generated procedurally by simulating a virtual environment populated by autonomous agents that play the roles of characters. The strong story approach advocates that a single 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.

We present an interactive narrative system that uses a combination of strong story and strong autonomy techniques to provide a story-based experience in which the player also has a high degree of agency. A prototype called IN-TALE (the Interactive Narrative Tacit Adaptive Leader Experience) demonstrates the combination of declarative story planning and reactive agents in the context of a story-based military leader training scenario. In the scenario, the user plays the role of a military leader in a foreign peacekeeping mission. Computer-controlled agents play the roles of merchants in a volatile marketplace. The scenario is based on dramatic principles and is designed to expose the user to dramatic and pedagogical situations. While the prototype scenario has an emphasis on training, we believe that the techniques described here have broad applicability to interactive entertainment applications.

The system is built on top of a 3D graphical computer game engine. Agent-controlled characters are capable of managing their own behaviors and interacting with the user through limited natural language. The player has a wide repertoire of actions, including dialogue acts, that he or she can perform at any time resulting in the look and feel of a social simulation instead of a more conventional branching interactive media. An intelligent component called the Automated Story Director maintains a script of expected events that it uses to provide high-level direction to the character agents to cause the story to progress. Character agents carry out local autonomous behaviors as well as behaviors that satisfy directives. While the character agents are capable of handling a wide degree of variability due to player interactivity, it is possible that the player can perform actions that make it impossible for the Automated Story Director to continue the story. In these circumstances, the Automated Story Director is capable of adapting the story structure to find new ways to coherently achieve its dramatic and pedagogical goals.

2 Related Work

Interactive narrative systems attempt to tell a story in which the user is able to make decisions and perform actions that dynamically affect the direction and/or outcome of the story. We catalog some related work according to the strong-autonomy/strong-story spectrum of technical approaches described in [9]. *Strong autonomous* approaches to interactive narrative are those in which narrative emerges through simulated interactions between believable character agents. Strong autonomy systems include [1], [3], and [5]. *Strong story* approaches to interactive narrative are those in which some form of global story control is implemented. Often story control is achieved through the use of a drama manager [4], a special agent that attempts to coerce the experience of the user to conform to a pre-existing story structure by directing or influencing the story world characters. Strong story systems include [20], [14], [21], and [17].

The work we present here is an attempt to integrate strong story and strong autonomy approaches to achieve greater impact on the user in terms of coherent narrative experience and the perception of user-agency. Systems that rely on combinations of autonomous agents and drama management include [2], [7], and [10]. Particularly relevant work are the *Mimesis* system [14; 21] and the *Façade* interactive drama [10]. Our system merges the generative narrative drama management approach pioneered by the *Mimesis* system combined with the believable agent capabilities of the ABL behavior specification language [11] used to endow NPCs with reactivity and believability of characters in *Façade*.

Also relevant are research efforts to create interactive narrative systems for training and education (c.f. [8], [16], [5], and [12]).

3 Example Plot

To motivate the problem, we present the following simple example narrative in which the user plays the role of a Captain in the U.S. Army as part of a peacekeeping mission deployed to a foreign country. The user plays the role of a Captain in charge of maintaining the security of civilian merchants and buyers in a marketplace.

The narrative is expected to unfold as follows. While the player is engaged in daily procedures concerning the new marketplace, a heated argument breaks out between two merchants. One merchant, named Saleh, accuses the other, named Hassan, of luring away his customers. Afterwards, Saleh approaches the trainee and complains that the presence of the peacekeeping troops is impinging on his ability to do business in the marketplace. He makes a dire prognostication that violence could ensue. Hassan, in contrast appears to be nothing but friendly. Later that day, Hassan slips away from the marketplace and returns concealing an improvised explosive device (presumably acquired from an insurgent conspirator). When Saleh steps away from his place of business, Hassan plants the bomb there. Shortly afterwards, the bomb goes off. Fortunately for all involved, the bomb is a dud, but the marketplace is nonetheless left in a state of chaos, panic, and confusion.

Note that if the player succeeds in preventing the attack by catching Hassan with the explosive device then it will be impossible for Hassan to plant the bomb and therefore impossible for the bomb to go off and establish the primary dramatic outcome. In the remainder of the paper, we describe a novel technique for trainee experience management that is designed incorporate strong story aspects into a system that retains the look and feel of a simulation.

4 Believable Characters

Social simulation is achieved through a collection of non-player characters (NPCs) that are reactive and appear intelligent, motivated, and reactive. 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." The objective is not to have agents that are competent reasoning agents, but agents that *appear* to be intelligent, motivated, emotional, and consequently believable. This emphasis on appearance is referred to as a "broad but shallow" approach to agents [2]. 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 Scenario Director (see Section 5). These scenario-specific behaviors are designed to modulate, mix with, and/or override local autonomous behaviors.

4.1 Agent Architecture

To achieve the desired life-like qualities we implemented our agents using the reactive planning language ABL (A Behavior Language) [11] using a behavioral infrastructure licensed from the Procedural Arts Behavior Library (PABL). The ABL language and PABL infrastructure were initially created for the interactive drama, *Façade* [10] and are 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 (see [6] for an enumeration of properties of believable agents).

In ABL, an activity (e.g., walking to the user, or speaking a line of dialog) is represented as a goal, and each goal is supplied with one or more behaviors to accomplish its task. An active goal chooses one of its behaviors to try. A behavior is a series of steps, that can occur sequentially or in parallel, that accomplish a goal. Preconditions are used to determine behavior applicability by matching against working memory elements (WMEs) that make up the agent's subjective knowledge about the world. A behavior may itself have one or more subgoals.

Further, to harness the dramatic power of multi-agent teams of characters, ABL supports authoring of joint goals and behaviors [11]. When a goal is marked as joint, ABL enforces coordinated entry into and exit from the team members' behaviors chosen to accomplish the goal. This coordination is transparent to the programmer and analogous to the STEAM multi-agent coordination framework [18]. The driving design goal of joint behaviors is to combine rich semantics for individual expressive behavior with support for the automatic synchronization of behavior across multiple agents.

4.2 Behavior Authoring

There are two broad categories of agent behaviors that must be authored: local autonomous behaviors and narrative directive 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 a loosely structured collections of sub-behaviors called "LAB goals", that depend on and assert simple events in episodic memory. For example, 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. A simple 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 (as described in Section 5). Further, user interaction afforded in NDBs usually needs to be richer and more responsive than in LABs. The collection of sub-behaviors that constitute an NDB are organized around the dramatic beat [9; 10], a component of the PABL infrastructure. A beat is a ~60-second-long dramatic interaction between characters such as a shared experience (e.g., witnessing a bombing), or a brief conflict about a topic (e.g., the user questioning an agent), or the revelation of an important secret. Beats are organized around a collection of "beat goal" behaviors, the dramatic content that the beat is designed to communicate to the user through animated performance.

The PABL authoring strategy for handling user interaction within a beat is to specify the "canonical" beat goal behavior logic (i.e., what dramatic performance the author intends the beat to accomplish), as well as a collection of beat-specific handler behaviors that modify this default logic in response to user interaction. Each interaction handler behavior is a demon that waits for some particular type of user interaction and "handles" it accordingly. User interaction includes dialogue interaction (the user can speak to the characters at any time by entering discourse acts, e.g. "disagree Saleh") and physical interaction (e.g. the user takes action such as "arrest Saleh"). Every NDB specifies some beat-specific handlers; additionally, there are more generic LAB handlers for handling interactions for which there are no beat-specific responses supplied by the current beat.

5 The Automated Story Director

There is a trade-off between narrative coherence and the trainee's perception of self-agency [14]. On one hand, a storytelling system will want to ensure a coherent progression of scenario events that lead the trainee through a dramatic and pedagogically relevant sequence of events. On the other hand, in a training simulation, the player needs to observe a realistic, populated social environment best delivered by a simulation and be able to problem-solve – to perform actions and make decisions. In our system, the player is allowed to perform any of a wide repertoire of communicative and physical actions at any time. Simulation with social agents alone, however, is not enough to ensure that the trainee is exposed to dramatic and pedagogically relevant situations in an appropriate and contextual order.

To ensure that the player's experience is managed and that the appropriate sequence of dramatic situations occur, an agent called the Automated Story Director acts as an unseen over-mind to coerce the player's experience to conform to a given scenario. Specifically, the Automated Story Director maintains a representation of the expected sequence of events that make up the scenario. From this representation, the story director derives and distributes directives to the NPCs to achieve certain conditions necessary to drive the scenario forward. For example, the Automated Story Director would direct the agent representing the Saleh character (from the example in Section 3) to establish the condition that the player distrusts the character – something that that agent might not choose to do if left to it's own devices.

Consistent with the design decision that the trainee can perform any of a wide repertoire of actions at any time, the Automated Story Director has a second responsibility: to monitor the simulation environment, detect inconsistencies between the simulation state and the expected narrative, and to reconcile any inconsistencies. This is essential in balancing the trade-off between narrative coherence and trainee self-agency because the player may perform actions that make it impossible for the story director to progress towards the desired dramatic situations. For example, the player could decide to apprehend Hassan before he plants the improvised explosive device in the marketplace. In this instance, the player has created an inconsistency between the simulation state (e.g. Hassan is detained) and the expected narrative representation (e.g. it must not be the case that Hassan is detained for the bomb to be planted). When inconsistencies arise, the narrative is adapted to reconcile the inconsistencies.

5.1 Anticipating Necessary Narrative Adaptations

Following [14; 21], 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 [19]. Plan steps, which in this application represent plot points, have 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. Other annotations, called *causal links*, are used to mark all causal relationships between the 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. A plan is not considered complete unless every precondition of every plan step is satisfied by a causal link.

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 trainee. We use a technique similar to that described in [14] to analyze the causal structure of the scenario to determine all possible inconsistencies between plan and simulation state that can occur during the entire duration of the scenario. For every possible inconsistency that can arise that threatens a causal link in the plan, an alternative scenario plan is generated. We have modified the original algorithm to use a tiered replanning approach. For each potential inconsistency that can arise, first the system attempts to repair the causal link that is threatened by the inconsistency. Barring that, the system attempts to remove any events that were dependent on the threatened causal link and then repair the plan by filling in events required to restore causal coherence. Finally, if all else fails, the system attempts to select new goals and relevant learning situations and rebuild the scenario plan.

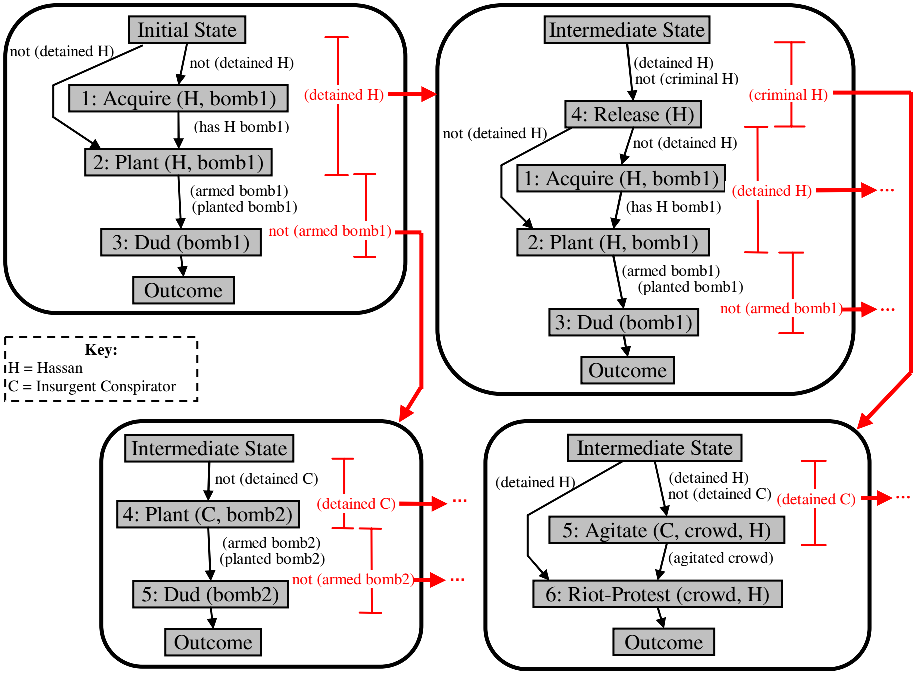

Fig. 1. Portion of the generated branching story

Narrative replanning can be performed offline to avoid delays due to computation [14]. 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 [14]. We find that by separating the system into a drama manager and autonomous character agents, the drama manager can focus on plot-level details and that relatively few user actions threaten the plot. This allows for a larger repertoire of user actions – including dialogue acts – to be implemented. Autonomous character agents are capable of providing a large amount of variability in the user's experience, especially with dialogue, without requiring the plot to be regenerated. We believe this will invoke in the user a greater sense of agency.

5.2 Example of Narrative Adaptation

Continuing the example story from Section 3, suppose the system is instantiated with a pre-constructed plan (shown in the upper left-hand corner of Fig. 1). Except for the plan steps and causal links in the original narrative plan, everything in the tree of contingency plans is automatically generated, including the potential inconsistency annotations of the original narrative itself. For the purposes of this discussion we have simplified the narrative further into the following steps:

1. `Hassan` acquires `bomb1`
2. `Hassan` plants `bomb1` in the marketplace
3. `bomb1` goes off as a dud.

The directed connections between steps are causal links indicating what must be true for a step to be applicable and which preceding step establishes that condition. This plan represents the narrative that will execute if the trainee does not inadvertently (or intentionally) cause an inconsistency between the simulation state and the plan structure. The plan is annotated with two intervals in which inconsistencies will threaten the causal coherence of the plan:

- `Hassan` is detained before he completes the planting of `bomb1`
- `bomb1` is disarmed before it goes off.

The former can occur if the trainee has Hassan arrested. The latter can occur if the player finds the bomb and calls in a bomb squad.

Each potential inconsistency annotation links to a contingency plan that repairs the inconsistency, should it occur. The detainment of Hassan links to a plan (upper right of Fig. 1) that is repaired by the addition of a step that releases Hassan from custody, ostensibly because he has not committed a crime. If the player searches Hassan and Hassan in fact has `bomb1` in his possession, he will be marked as a criminal. This does not cause an inconsistency that threatens the original plan. But once detained, if Hassan is marked as a criminal, he cannot be released, instantly causing a second transition to the plan in the lower right of Fig. 1.

The plan in the lower left of Fig. 1 represents the narrative in which `bomb1` is found and disarmed by the trainee. In this case, the outcome of the original narrative cannot be achieved. This plan repairs the original narrative by having an insurgent conspirator step in and complete the attack on the marketplace with a second bomb.

The plan in the lower right of Fig. 1 represents the narrative in which Hassan is caught with a bomb and permanently detained. Like the previously described contingency, the outcome of the original narrative can still be achieved by having an insurgent conspirator complete the attack with a second bomb. However the planner can find a more appealing variation. The insurgent conspirator uses the detention of Hassan to agitate a crowd. The crowd then riots in protest of the player's actions.

6 Integrating Story Direction and Believable Character Agents

Believable character agents, the Automated Story Director, and a variant of the example scenario from Section 3 have been combined into a prototype called IN-TALE (the Interactive Narrative Tacit Adaptive Leader Experience). The system is built on top of a 3D computer game engine. The system is currently in prototype stage, implementing a version of the example scenario in Section 3 that can be automatically adapted in a large number of ways. The Automated Story Director generates a tree of contingency plans with over 1000 paths, although most are minor variations for maintaining narrative coherence. Fig. 2 shows a screenshot of the trainee (central avatar) being confronted by an NPC. A set of believable social agents are implemented as semi-autonomous, intelligent agents in ABL. Each ABL agent controls a single virtual avatar in the game engine. The Automated Story Director receives state updates from the game engine via an interface described in [13].

As in [1], the Automated Story Director is responsible for coercing the player's experience to conform to a high-level plot structure. A high-level plot is one that

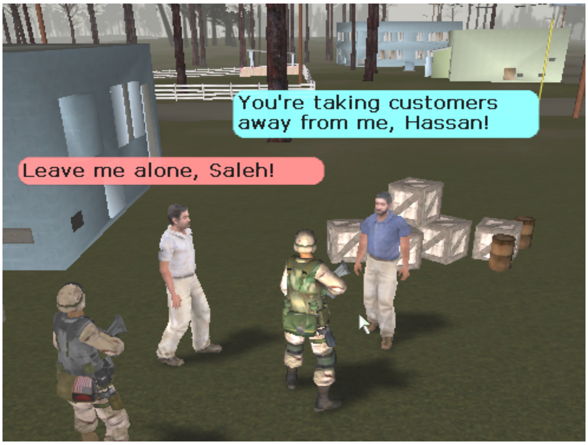

Fig. 2. Screenshot of trainee (central) confronting a non-player character

describes the narrative progression in terms of a sequence of world states – or *plot points* – that describe situations or scenes that should occur without reference to the primitive character actions that have occurred to bring out each state or the primitive character actions that should occur once a situation occurs. The directives from the Automated Story Director to the NPCs are not detailed instructions. Directives are goals the NPC must adopt, in terms of declarative world state change. Character agents are left to determine the best way to achieve the directives, barring any behavior case world states explicitly prohibited by the Director. This gives the agents the leeway to engage in behaviors that are believable, achieve scenario goals, and take full advantage of the current world situation, e.g. executing one or more NDBs as described in Section 4.

The paradigm of high-level directives and low-level agent autonomy opens up the possibility of an agent selecting joint behaviors. A joint behavior is a method – one of many methods known to the agent for achieving a goal – that co-opts the participation of other NPCs in the world for close coordination of activity and/or dramatic effect. For example, if the agent representing the Hassan character is directed to acquire an explosive device (e.g. `(has Hassan bomb1)`), that character might not be able to do so while the trainee is in close proximity. Assume there is a directive that `(knows player (has Hassan bomb1))` should never become true. The agent may select a joint behavior – one of many known methods for acquiring an object – that in conjunction with another agent, possibly a bystander, creates a diversion. The joint behavior provides coordination so that Hassan knows to slip off while the player is distracted.

To avoid the appearance of agents that schizophrenically [15] switch between goals– in this case goals autonomously selected for believability and goals demanded by the Story Director – behaviors selected to achieve the appearance of believability (LABs) and behaviors that achieve narrative goals (NDBs) must mix seamlessly. Local autonomous behaviors can run in parallel and/or interleave with behaviors selected to achieve scenario goals. “Real life” behaviors can be modulated to believably blend with the high-level scenario behaviors imposed on them. Modulations of LABs include: timing alteration to accommodate the needs of the scenario; reducing the number of physical resources required to avoid conflicts with scenario-driven behaviors; and avoiding actions that would violate overall believability in any way.

Mixing NDBs with LABs involves annotating NDBs and LABs with the resources they require: location requirements, object requirements, emotional state requirements and so on. When an NDB prepares for execution in order to fulfill a directive from the Automated Story Director, the NDB announces its resource requirements. LABs are coded with behavior variations in order to gracefully degrade their performance to accommodate the needs of the more important NDBs, while still behaving believably. When a LAB is required to interrupt or even abort its execution to serve the NDB’s needs, it selects from a variety of short transition-out sub-behaviors to believably “glue”, i.e., explain why. For example, if a *cleaning the store* LAB needs to be truncated or aborted in order for the agent to participate in NDBs to acquire and plant a bomb, the agent may choose to insert some dialogue to the effect of, “Hmmm, the store is pretty clean today… I think sweeping can wait till tomorrow”. Similarly, whatever LAB begins after the NDB ends can select from transition-in sub-behaviors similarly “gluing” the agent’s behavior back into its daily routine.

7 Conclusions

Our approach draws heavily from previous research. A generative narrative approach is used by our drama manager, the Automated Story Director, modeled after the methods developed in [14; 21]. The believable agents are based on the ABL behavior specification language [11] originally developed for the *Façade* [10] interactive drama system. Our goal is to provide an interactive narrative experience that merges the benefits of the strong story approach with the benefits of the strong autonomy approach. Interactive narrative requires two seemingly conflicting requirements: coherent narrative and user agency. On one hand, we would like to ensure a coherent narrative experience for a player that always has a dramatic quality that leads to a recognizable outcome. One the other hand, we would like the player to be able to perform actions to effect change in the world in a non-superficial way – to be able to influence the direction and outcome of the narrative. Our system presents the user of the perception of a simulation-like experience to afford greater agency to the user. Meanwhile, plot-level drama management ensures that the simulation progresses in such a way as to provide the user with a narrative experience. Specifically, we provide for believable agents that can interact with the player in real-time and are autonomous enough to plan their own courses of action for increased believability. The agents are directed by the Automated Story Director at a high level of abstraction so that local situations can play out naturally. Most player interaction is handled by the believable agents. When the player performs actions that change the world in a way that makes it impossible for the Story Director to continue the plot, the Automated Story Director adapts the plot to incorporate the player's actions while still achieving the dramatic goals.

Acknowledgements

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

References

[1] Aylet,, R.: Emergent narrative, social immersion and "storification." *Proc. of the 1st International Workshop on Narrative and Interactive Learning Environments* (2000).

[2] Bates, J.: Virtual Reality, Art, and Entertainment. *Presence: The Journal of Teleoperators and Virtual Environments*, 1, (1992).

[3] Cavazza, M., Charles, F., & Mead, S.: Planning Characters' Behaviour in Interactive Storytelling. *Journal of Visualization and Computer Animation*, 13 (2002) 121-131.

[4] Kelso, M., Weyhrauch, P., Bates, J.: Dramatic Presence. *Presence: The Journal of Teleoperators and Virtual Environments*, 2 (2005).

[5] Louchart, S., Aylett, R., Dias J., & Paiva, A.: Unscripted narrative for affectively driven characters. *Proc. of the 1st National Conf. on Artificial Intelligence and Interactive Digital Entertainment* (2005).

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

[7] Magerko, B. Story Representation and Interactive Drama. *Proc. of the 1st Conf. on Artificial Intelligence and Interactive Digital Entertainment* (2005).

[8] Marsella, S., Johnson, W.L., LaBore, K.: Interactive Pedagogical Drama. *Proc. of the 4th Int. Conf. on Autonomous Agents* (2000).

[9] Mateas, M., & Stern, A.: Towards Integrating Plot and Character for Interactive Drama. *AAAI Spring Symposium on Social Intelligent Agents: The Human in the Loop* (2000).

[10] Mateas, M, & Stern, A.: *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).

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

[12] Mott, B.W., Lester, J.C.: U-Director: A Decision-Theoretic Narrative Planning Architecture for Storytelling Environments. *Proc. of the 5th Int. Joint Conf. on Autonomous Agents and Multi Agent Systems* (2006).

[13] Riedl, M.O.: Towards Integrating AI Story Controllers and Game Engines: Reconciling World State Representations. *Proc. of the IJCAI Workshop on Reasoning, Representation and Learning in Computer Games* (2005).

[14] Riedl, M.O, Saretto, C.J., and Young, R.M.: Managing Interaction Between Users and Agents in a Multi-Agent Storytelling Environment. *Proc. of the 2nd Int. Conf. on Autonomous Agents and Multi Agent Systems* (2003).

[15] Sengers, P.: Schizophrenia and Narrative in Artificial Agents. In: M. Mateas & P. Sengers (eds.): *Narrative Intelligence*. John Benjamins (2003).

[16] Si, M., Marsella, S., & Pynadath, D.V.: Thespian: Using multi-agent fitting to craft interactive drama. *Proc. of the 4th Int. Joint Conf. on Autonomous Agents and Multi Agent Systems* (2005).

[17] Szilas, N.: IDtension: a narrative engine for interactive drama. *Proc. of the 1st Int. Conf. on Technologies for Interactive Digital Storytelling and Entertainment* (2003).

[18] Tambe, M.: Towards Flexible Teamwork. *Journal of Artificial Intelligence Research*, 7 (2003).

[19] Weld, D.: An Introduction to Least Commitment Planning. *AI Magazine*, 15 (1994).

[20] Weyhrauch, P.: *Guiding Interactive Fiction.* Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University (1997).

[21] Young, R.M., Riedl, M.O., Branly, M., Jhala, A., Martin, R.J., and Saretto, C.J.: An Architecture for Integrating Plan-Based Behavior Generation with Interactive Game Environments. *Journal of Game Development*, 1 (2004) 51-70.