42

Techniques for AI-Driven Experience Management in Interactive Narratives

Vadim Bulitko, Greg Lee, Sergio Poo Hernandez, Alejandro Ramirez, and David Thue

42.1 Introduction

Recently, there has been a trend to include open-world gameplay in story-rich video games (e.g., [Petit 14]). Giving the player agency can lead to his or her affecting the original story in a way that the game designers may not have anticipated. To illustrate, imagine a game version of the *Little Red Riding Hood* story [Perrault 1697, Grimm 1812]. Suppose that the player, controlling Red, kills the wolf early in the game. The rest of the story is then broken, since the wolf is unable to eat the grandmother.

This chapter describes several artificial intelligence (AI) techniques that can help game designers provide players with more agency while still maintaining a desired narrative feel for their games. In the example of Red killing the wolf, the AI techniques can automatically revise the story upon the wolf's demise (e.g., another predator appears) so that certain authorial goals are maintained (e.g., the grandmother is eaten).

42.2 AI-Driven Experience Management

The task we consider in this chapter is that of managing a player's experience, driven by AI. This is an expansive area in which several approaches have been proposed, and it offers several avenues to explore [Riedl 13]. The key idea is to create an AI game master (AI GM), which is bundled with every copy of the game. As the player proceeds through the narrative, making narrative choices, the AI GM monitors the player's actions and dynamically modifies the story in a way that the story designer would have wanted. In this chapter, we use the terms *story designer* and *story author* interchangeably, and this view of interactive storytelling is also known as *delayed authoring* [Thue 08].

For the sake of consistency with existing literature, we also use the terms *AI GM* and *AI experience manager* interchangeably throughout the chapter. Computationally, we view experience management as an optimization problem where the AI GM attempts to optimize the player's experience according to a designer's specified objectives [Thue 12]. We detail our approach in the following sections.

42.2.1 Player in a Game Environment

If a game narrative is going to be modified by AI, it first needs to be represented in a computer-readable format. This means that the narrative state (e.g., Red just met the wolf) and all actions available to change the state (e.g., shoot the wolf or reason with the wolf) need to be formally described. While numerous methodologies have been used to represent game worlds in an AI-readable format, in this article, we will use the Planning Domain Definition Language (PDDL) [McDermott 98]. Listing 42.1 illustrates the PDDL encoding of the action *eat*, which one game character may do to another in the *Little Red Riding Hood* story.

Note that the action is represented in a programming-like notation with variables (e.g., `?eater`) and predicates (e.g., `alive`). It says that the `eater` can eat the `eatee` if the `eater` is a predator, the `eatee` is a person, the `eater` knows the `eatee`, and the `eater` is hungry. They both must be alive and the `eatee` cannot yet be eaten. Once the action is applied, the `eatee` is eaten and is located inside the `eater`, and the eater is no longer hungry. Table 42.1 illustrates an application of this action.

Listing 42.1. An action template that can be instantiated into an in-game action.

```
(define (action eat)
    :parameters (?eater ?eatee)
    :precondition (and (knows ?eater ?eatee)
                       (predator ?eater)
                       (alive ?eatee)
                       (alive ?eater)
                       (not (eaten ?eatee))
                       (hungry ?eater)
                       (person ?eatee))
    :effect (and (eaten ?eatee)
                       (in ?eatee ?eater)
                       (not (hungry?eater))))
```

Table 42.1 Changes in Game Narrative State Caused by an Action

Current State	Action	New State
`(person red)`		`(person red)`
`(alive red)`	`(eat wolf red)`	`(alive red)`
`(predator wolf)`		`(predator wolf)`
`(alive wolf)`		`(alive wolf)`
`(hungry wolf)`		***`(not (hungry wolf))`***
`(knows wolf red)`		`(knows wolf red)`
`(knows red wolf)`		`(knows red wolf)`
		`(in red wolf)`
		`(eaten red)`

In this example, the action's variables `eater` and `eatee` are bound to the wolf and Red, respectively. The current state is shown in the first column of Table 42.1. It is represented with a series of facts (e.g., Red is a person and the wolf is a predator) and satisfies the preconditions of the action. Once the action is applied in the current state, the new state is generated with the new facts shown in bold.

42.2.2 AI Manager Modifying Narrative

Once the narrative state and the player's actions are represented in a computer-readable format, the AI can reason over the narrative and modify the game world in response to the player's actions on the fly. To do so, the AI GM needs to know what the author wants to happen in the story.

Traditionally, games encode nonlinear narrative as story graphs (or story trees). The game engine selects the appropriate branch based on story tokens (e.g., whether the player saved Ashley or Kaiden in *Mass Effect* [BioWare 07]). An alternate approach to specifying the author's wishes is through PDDL-described game facts. Figure 42.1 depicts a skeleton of the interactive narrative for the *Little Red Riding Hood* [Ramirez 13] where the story author has two goals: (1) a predator should eat both Red and her grandmother, and (2) Red should give her grandmother a cake (shaded nodes in the story graph on the left).

When the player kills the wolf, both `(eat predator red)` and `(eat predator granny)` are invalidated as the wolf is dead and unable to eat anyone. The AI manager can then engage an automated planner to satisfy these goals from the current state. Given the action templates that are available, the planner may come up with two stories. In one, another predator (named Grendel) greets the player and takes over from the deceased wolf. In the other, a magic fairy resurrects the original wolf who can then resume his eating duties.

42.2.3 Choosing between Stories

Whenever several stories are available that satisfy the authorial goals, the AI GM needs to decide which one to run. To choose among them, the game master can rely on the author's specification of an ideal player experience. For instance, suppose that the author wants to maximize the player's enjoyment of the story. Suppose also that on the basis of the player's previous actions, the AI experience manager believes the player to have an inclination

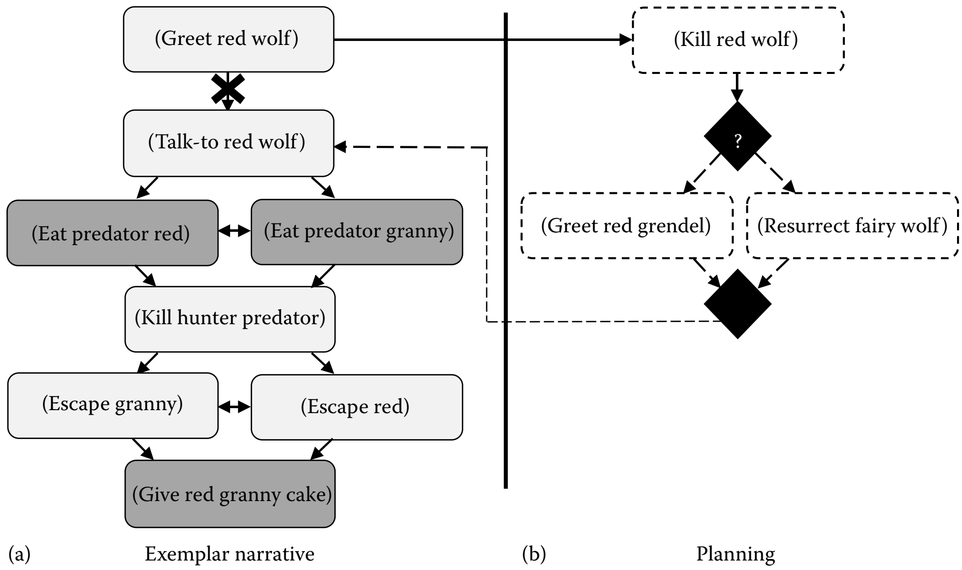

Figure 42.1

Author's desired events (a) and AI experience manager's actions (b). The arrows show possible temporal relations between the events.

toward combat. Then in deciding between two possible narratives in Figure 42.1, the manager may decide that introducing Grendel—effectively a bigger, meaner version of the original wolf—is a more suitable option than presenting a resurrecting fairy. The manager then runs the narrative content with Grendel until the player takes the next action and once again prompts the manager to modify the game world. We give examples of how an author can describe an ideal player experience in the following sections.

42.3 AI-Driven Experience Management: Common Techniques

The previous section introduced AI-driven player experience management. The interactive narrative was encoded in a computer-readable format, story branches were either preauthored or automatically planned out, and a specification of the player's ideal experience was used to select among them. Figure 42.2 shows how these steps are related to gameplay, with the AI GM monitoring and updating the game to manage what happens next.

In this section, we detail computational techniques that power those steps. In doing so, we provide the game designer with a set of tools that he or she can mix and match to suit the needs of his or her particular game.

42.3.1 Technique: Narrative Generation

Given a formal representation of the story world and a history of the player's actions, the AI GM can automatically compute possible narratives that (1) are consistent with the player's actions and (2) satisfy any future authorial goals. One approach to doing so is to use an automated planner. In the past, narrative-centric planners have been used

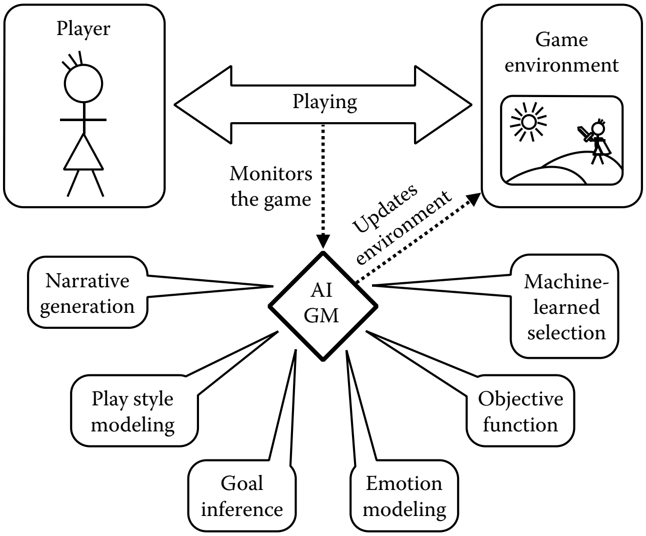

Figure 42.2

A high-level view of AI experience management.

(e.g., *Longbow* [Young 94]). More recent systems attempt to use general-purpose off-the-shelf planners such as *Fast Downward* [Helmert 06].

Since a planner cannot produce the actual high-quality writing or multimedia content to go with the narrative stubs generated in PDDL, the corresponding narrative pieces need to be pregenerated during game development and then fleshed out by human writers and artists. Then, during the game, the AI planner is effectively *selecting* from a library of narrative fragments in order to assemble start-to-finish narratives on the fly while satisfying the world consistency (e.g., the wolf is dead) and authorial constraints (e.g., the grandmother needs to be eaten).

42.3.2 Technique: Play Style Modeling

Knowing something about the player can be helpful when choosing between narratives. One approach is to model the player as a vector of numbers, each number indicating the player's inclination toward a certain play style. The play styles can be taken from player archetypes such as the canonical RPG types [Laws 01]. For instance, the vector could be (F: 0.9, M: 0.2, S: 0.1, T: 0.4, P: 0.3), which indicates that the player has often played as a fighter (0.9) but less so as a method actor (0.2), storyteller (0.1), tactician (0.4), or power gamer (0.3). These values are maintained by the AI GM when it observes the player's actions in game [Thue 07]. A simple way to implement this is to annotate each action template (e.g., Listing 42.1) with a vector of updates to the model. For instance, whenever the player kills anyone, the model's estimate of his or her fighter inclination could increase by 0.3.

The player's play style inclinations may change over time. Thus, the model should track the player's current/recent inclinations. This can be handled by gradually moving all vector components toward their neutral values over time.

42.3.3 Technique: Goal Inference

Knowing something about the player's inclinations toward different play styles can help the AI manager infer the player's current goals. For instance, if the player controlling Red has previously shown an inclination toward fighting, it is likely that he or she will try to kill Grendel.

We can implement this type of inference by having the game designer encode potential player goals in PDDL and specify a correlation between play styles and goals. To illustrate, suppose that when Grendel is first introduced, any player may intend to either kill Grendel (goal #1) or avoid him (goal #2). Then, the correlation can be specified as in Table 42.2.

This means that someone purely interested in fighting would have a strong (0.9 out of 1) intention to kill Grendel and a weak intention (0.1) to avoid him. In practice, a player is usually inclined toward a mix of different play styles. Thus, the goal intentions of a given player are computed as a matrix product of the correlation matrix and the player model (F: 0.9, M: 0.2, S: 0.1, T: 0.4, P: 0.3):

$$\begin{pmatrix} 0.9 & 0.7 & 0.2 & 0.4 & 0.6 \\ 0.1 & 0.3 & 0.6 & 0.8 & 0.1 \end{pmatrix} \times \begin{pmatrix} 0.9 \\ 0.2 \\ 0.1 \\ 0.4 \\ 0.3 \end{pmatrix} \approx \begin{pmatrix} 1.31 \\ 0.56 \end{pmatrix} \tag{42.1}$$

Note that both the correlation matrix and the player model shown in Equation 42.1 are transposed to make the matrix product work. The result shows us the player represented by our current player model has a weight of 1.31 to fight Grendel and a weight of 0.56 to avoid it. We also normalize the vector to keep all intention values between 0 and 1. The normalized result is approximately (0.7, 0.3), indicating that this player is likely intending to kill (and not avoid) Grendel [Poo Hernandez 14].

42.3.4 Technique: Emotional Modeling

One way for the game designers to specify an ideal player experience is to give the desired emotional arc. For instance, many games follow the classic Aristotelian tension arc [BioWare 07, BioWare 09, Bethesda 11], while some experiment with periodic functions. A commercially deployed example of the latter is Valve's *Left 4 Dead*, which uses an AI Director to keep the player on a sinusoidal tension curve by dynamically shaping the zombie influx [Booth 09].

Table 42.2 Play Style and Goal Correlation Example

Play Style Inclination	Goal 1: Kill Grendel	Goal 2: Avoid Grendel
Fighter	0.9	0.1
Method actor	0.7	0.3
Storyteller	0.2	0.6
Tactician	0.4	0.8
Power gamer	0.6	0.1

While *Left 4 Dead's* AI Director models the player along a single emotional dimension (tension), the appraisal-style models of emotions [Lazarus 91] are able to infer the player's emotions in several dimensions [Ortony 90]. For example, CEMA [Bulitko 08] is a lightweight subset of EMA [Marsella 03] that focuses on just four emotions: joy, hope, fear, and distress. Thus, the player's emotional state is represented with four numbers, each expressing the current intensity of the corresponding emotion. Thus, (J:0.8, H:0.6, F:0.2, D:0) would represent a player who is presently joyful (0.8) and hopeful (0.6) but slightly fearful (0.2) and not distressed (0).

To compute the numbers, an appraisal-style model needs to know the player's goals and the likelihood of accomplishing them. To derive the player's goals, one can use the technique of goal inference presented earlier in the article. The likelihood of accomplishing the goals can be computed based on the player's character attributes (e.g., a stealthy character will probably be able to avoid Grendel).

To illustrate, consider a player whose goal intentions to kill Grendel and avoid dying are computed as (0.7, −0.3). Suppose this player has a 50% chance of killing Grendel and a 10% chance of dying. A desirable but uncertain goal elicits hope and hence the system models the player as being hopeful at the intensity of $0.5 \times 0.7 = 0.35$. According to appraisal-style models, once a desirable goal is certain, it no longer elicits hope; instead, it elicits joy. Since the goal of killing Grendel is uncertain, there is no joy from it (yet). Dying is undesirable to this player (values for undesirable outcomes are expressed by negative numbers, −0.3 in this example), and hence it elicits the emotion of fear with an intensity of $0.1 \times 0.3 = 0.03$. There is no distress yet since the player is not certain to die. Hence, the player's emotional state is computed as (J:0, H:0.35, F:0.03, D:0) [Gratch 06].

42.3.5 Technique: Objective Function Maximization

Player inclinations and emotional state models can be used by the AI GM to select among alternative narratives, with the goal of achieving the author's notion of an ideal player experience. A simple way of doing so is to annotate each narrative with its suitability with respect to different styles of play. For example, introducing Grendel may be most suitable for a fighter, whereas a resurrecting fairy may appeal to a storyteller. Computationally, the suitability of a narrative is a vector of numbers, one for each style of play. For instance, a simple annotation for the *introduce Grendel* narrative can be (F: 0.9, M: 0, S: 0, T: 0, P: 0), whereas the *introduce magic fairy* narrative can be annotated as (F: 0, M: 0, S: 0.9, T: 0, P: 0). Given several annotated alternative narratives, the manager can take a dot product between the player inclination model and each annotation. The narrative with the highest dot product is then presented to the player [Thue 07, Ramirez 13]. In our example, the player's play style inclinations are modeled as (F: 0.9, M: 0.2, S: 0.1, T: 0.4, P: 0.3), which means that the dot product for *introduce Grendel* is 0.81 (Equation 42.2), whereas the dot product for *introduce magic fairy* is 0.09 (Equation 42.3). Hence, our player will face Grendel after killing the wolf:

$$\begin{pmatrix}0.9 & 0 & 0 & 0 & 0\end{pmatrix} \cdot \begin{pmatrix}0.9 & 0.2 & 0.1 & 0.4 & 0.3\end{pmatrix} = 0.81 \tag{42.2}$$

$$\begin{pmatrix}0 & 0 & 0.9 & 0 & 0\end{pmatrix} \cdot \begin{pmatrix}0.9 & 0.2 & 0.1 & 0.4 & 0.3\end{pmatrix} = 0.09 \tag{42.3}$$

A more advanced approach is to use the play style inclinations to infer the player's intentions. Those can then be used to predict how the player's emotional state will be affected by each of the alternative narratives. The narrative that manages to bring the player's emotional state the closest to the target emotional state (as specified by the author) is then presented to the player. To illustrate, suppose the author wants their players to experience the following emotional state (J:0, H:0.4, F:0.03, D:0) at this point in the story. The narrative introducing Grendel is predicted to put the player in the emotional state (J:0.8, H:0.6, F:0.2, D:0), whereas the narrative introducing the magic fairy is expected to elicit the emotional state (J:0, H:0.35, F:0.03, D:0). The Euclidean distance between the expected emotional state (J:0.8, H:0.6, F:0.2, D:0) and the target emotional state (J:0, H:0.4, F:0.03, D:0) is 0.83:

$$\sqrt{(0.8-0)^2+(0.6-0.4)^2+(0.2-0.03)^2+(0-0)^2} \approx 0.83 \tag{42.4}$$

On the other hand, the distance between the emotional state (J:0, H:0.35, F:0.03, D:0) and the target emotional state (J:0, H:0.4, F:0.03, D:0) is only 0.05. The AI manager tries to minimize the distance and thus will select the narrative with the magic fairy.

42.3.6 Technique: Machine-Learned Narrative Selection

Yet another alternative for selecting narratives is to use machine learning during the development stage to automatically acquire a mapping from game and player states to the set of alternative narratives. One such mapping is a ranking function, similar to how Internet search engines map user queries to a ranked list of web pages. This approach is appropriate when training data are available. For instance, if human game masters could be observed and their narrative selection recorded, then the resulting corpus of data could be used to machine-learn an AI approximation to a human game master. We describe an implementation of this approach in the following section.

42.4 Implementations

Over the last seven years, our research group at the University of Alberta has implemented AI experience managers using different combinations of the techniques given earlier. In the following sections, we briefly review these implementations and describe how they were tested.

42.4.1 PaSSAGE

Player-Specific Stories via Automatically Generated Events (PaSSAGE) [Thue 07, Thue 11] is an AI experience manager that combines two techniques: (1) play style inclination modeling and (2) maximizing a simpler version of the aforementioned objective function. We deployed PaSSAGE in the domain of RPG-style games, using off-the-shelf game engine technology from both *Neverwinter Nights* [BioWare 02] and *Dragon Age: Origins* [BioWare 09]. Our first test bed, *Annara's Tale*, is an adaptation of the *Little Red Riding Hood* that included five different endings, eight substantially different paths through the story, and three points at which PaSSAGE could choose between story branches. Our second test bed, *Lord of the Borderlands*, is an original story that included 16 different endings, 32 substantially different paths through the story, and two points at which PaSSAGE could influence what happened next. When compared to a uniform random manager (which still obeyed the game's authorial constraints), PaSSAGE scored higher with respect to player-reported fun with high confidence (as tested with 133 undergraduate students). While preliminary, this result shows promise for applying PaSSAGE in both current and next generation games.

42.4.2 PAST

Player-Specific Automated Storytelling (PAST) [Ramirez 12, Ramirez 13] added narrative generation to PaSSAGE. Specifically, instead of hand-scripting the narrative continuations (as was done in PaSSAGE), PAST used the automated planning module from the Automated Story Director [Riedl 08] to compute narratives that were consistent with the player's actions and the authorial goals.

We evaluated PAST with a text-based choose-your-own-adventure version of the *Little Red Riding Hood* story. Within each game, the player made four consecutive narrative choices. For each player choice, PAST computed several alternative accommodations (i.e., alternate stories based on a virtual domain description aligned with authorial goals) and selected one using a play style model similar to the one in PaSSAGE. More than 30 different narrative trajectories were available to the players. Players' self-reported perception of fun and agency indicated positive trends [Ramirez 13].

42.4.3 PACE

Player Appraisal Controlling Emotions (PACE) [Poo Hernandez 14] is our latest AI experience manager. It uses the four techniques of narrative generation, play style modeling, goal inference, emotion modeling, and the second, more advanced type of the objective function. Unlike PAST, we used a PDDL-compatible off-the-shelf automated planner to compute possible narratives. PACE uses the full objective function to predict the player's emotional state along each of the alternative narratives. It selects the narrative that brings the player closest to the target emotional trajectory.

PACE is currently being deployed within the novel video game *iGiselle*: an interactive reimaging of the classic Romantic ballet *Giselle* [Gautier 41]. In *iGiselle*, the player assumes various ballet poses (read by Microsoft Kinect) to control their avatar and make narrative choices. The narrative is presented via a combination of still images, prerecorded voiceovers, videos, and music.

42.4.4 SCoReS

Sports Commentary Recommendation System (SCoReS) chooses stories for sports broadcasts [Lee 13]. It uses machine-learned narrative selection to pick the piece of color commentary most suitable to the state of a sports game at any given moment. It was evaluated in the domain of baseball and received positive feedback from both the users and human expert color commentators.

42.5 Conclusion and Future Work

We have presented the task of managing a player's narrative experience in a video game. In doing so, we decomposed the problem into several steps and presented specific techniques that can be used to implement an AI GM for a particular video game. We then briefly described four such game masters that we have developed in our research group.

Future work will continue implementing and testing these types of approaches within different games. We also hope to build a narrative space exploration tool that will combine the narrative generation technique with player models data-mined from existing game telemetry. Game designers could then use this tool to interactively explore and shape a narrative space in the early stages of a game's story development.

References

[Bethesda 11] Bethesda. 2011. The Elder Scrolls V: Skyrim. http://www.elderscrolls.com/skyrim.

[BioWare 02] BioWare. 2002. Neverwinter Nights. http://www.bioware.com/en/games/.

[BioWare 07] BioWare. 2007. Mass Effect. http://www.bioware.com/en/games/.

[BioWare 09] BioWare. 2009. Dragon Age: Origins. http://www.bioware.com/en/games/.

[Booth 09] Booth, M. 2009. The AI systems of Left4Dead. Keynote. *Fifth Artificial Intelligence and Interactive Digital Entertainment Conference*.

[Bulitko 08] Bulitko, V. et al. 2008. Modeling culturally and emotionally affected behavior. *Proceedings of Artificial Intelligence and Interactive Digital Entertainment Conference*, pp. 10–15. AAAI Press.

[Gautier 41] Gautier, T. et al. 1841. *Giselle*.

[Gratch 06] Gratch, J. and S. Marsella. 2006. Evaluating a computational model of emotion. *Journal of Autonomous Agents and Multiagent Systems* 11:23–43.

[Grimm 12] Grimm, J. and W. Grimm. 1812. Little red cap. In *Kinderund Hausmärchen*, 1st edn., Vol. 1.

[Helmert 06] Helmert, M. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research* 26:191–246.

[Laws 01] Laws, R. D. 2001. *Robin's Laws for Good Game Mastering*. Austin, TX: Steve Jackson Games.

[Lazarus 91] Lazarus, R. S. 1991. *Emotion and Adaptation*. New York: Oxford University Press.

[Lee 13] Lee, G., V. Bulitko, and E. Ludvig. 2013. Automated story selection for color commentary in sports. *IEEE Transactions on Computational Intelligence and AI in Games* 6(2):144–155.

[Marsella 03] Marsella, S. and J. Gratch. 2003. Modeling coping behavior in virtual humans: Don't worry, be happy. *Proceedings of Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 313–320.

[McDermott 98] McDermott, D. et al. 1998. PDDL-the planning domain definition language.

[Ortony 90] Ortony, A., G. L. Clore, and A. Collins. 1990. *The Cognitive Structure of Emotions*. Cambridge, U.K.: Cambridge University Press.

[Perrault 97] Perrault, C. 1697. Little Red Riding Hood.

[Petit 14] Petit, C. 2014. Balancing freedom and story in *Dragon Age: Inquisition*. Gamespot. http://www.gamespot.com/articles/balancing-freedom-and-story-in-dragon-age-inquisition/1100–6419121/ (accessed May 4, 2014).

[Poo Hernandez 14] Poo Hernandez, S., V. Bulitko, and E. St. Hilaire. 2014. Storytelling with artificial intelligence. *Proceedings of the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment,* North Carolina State University, Raleigh, NC, p. 7.

[Ramirez 12] Ramirez, A. and V. Bulitko. 2012. Telling interactive player-specific stories and planning for it: ASD + PaSSAGE = PAST. *Proceedings of the Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment,* Stanford University, Palo Alto, CA, pp. 173–178. AAAI Press.

[Ramirez 13] Ramirez, A., V. Bulitko, and M. Spetch. 2013. Evaluating planning-based experience managers for agency and fun in text-based interactive narrative. *Proceedings of the Ninth AAAI Conference on AI and Interactive Digital Entertainment,* Noreastern University, Boston, MA, pp. 65–71. AAAI Press.

[Ramirez 14] Ramirez, A. and V. Bulitko. 2014. Automated planning and player modelling for interactive storytelling. *IEEE Transactions in Computational Intelligence and AI in Games* 12, in press.

[Riedl 08] Riedl, M. et al. 2008. Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on Systems Science and Applications* 4(2):23–42.

[Riedl 13] Riedl, M. and V. Bulitko. 2013. Interactive narrative: an intelligent systems approach. *AI Magazine* 34(1):67–77.

[Thue 07] Thue, D. et al. 2007. Interactive storytelling: A player modelling approach. *Proceedings of the Third AAAI Conference on AI and Interactive Digital Entertainment*, Stanford, CA, pp. 43–48. AAAI Press.

[Thue 08] Thue, D. et al. 2008. Making stories player-specific: Delayed authoring in interactive storytelling. *Proceedings of the First Joint International Conference on Interactive Digital Storytelling*, pp. 230–241. Springer, Berlin, Germany.

[Thue 11] Thue, D. et al. 2011. A computational model of perceived agency in video games. *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Stanford, CA, pp. 91–96. AAAI Press.

[Thue 12] Thue, D. and V. Bulitko. 2012. Procedural game adaptation: Framing experience management as changing an MDP. *Proceedings of the Fifth Workshop in Intelligent Narrative Technologie*, Stanford, CA. AAAI Press.

[Young 94] Young, M. R. 1994. A developer's guide to Longbow discourse planning system. University of Pittsburgh Intelligent Systems Program Technical Report 94-4, Pittsburgh, PA.