

Telling Interactive Player-Specific Stories and Planning for It: ASD + PaSSAGE = PAST

Alejandro Ramirez and Vadim Bulitko

Department of Computing Science, University of Alberta
Edmonton, Alberta, T6G 2E8, Canada
{ramirez|bulitko}@cs.ualberta.ca

Abstract

Artificial Intelligence (AI) techniques have been used to improve overall player satisfaction in video games. A thriving area of such research is interactive AI-assisted story-telling. One such line of research explored gains of automatically fitting the story to each individual player during the game via player modeling. Another research line used AI planning techniques to create contingency stories at design time. In this paper we propose a principled way to combine these two lines of research. We describe a system that uses a player model both at the design time and at the play time to generate and select stories fitting a specific player. We implement an early prototype and present it in the paper.

Introduction

From Shakespeare’s “Romeo and Juliet” to George Lucas’ “Star Wars” to BioWare’s “Jade Empire” to camp-fire stories to baseball commentary, story-telling is a fundamental part of entertainment. A strong narrative resonates with our minds, hearts and souls and keeps us engaged. We remember the stories of our childhood and re-tell them to our own children. Story-telling has delighted and saddened the human race since the beginning of time and shows no signs of slowing down. But can it be improved with technology?

There are two areas that can be improved in the traditional fixed narrative (e.g., literature and movies). First, there is evidence that one size does not fit all and thus higher player enjoyment can be had by customizing the narrative to each individual audience member. This requires knowing something about the audience as the narrative is presented to them. Second, people appear to enjoy having an influence on the narrative being told. Such agency, the ability to change world or environment (Thue et al. 2011), can be difficult to achieve as it may go against the story the human author wishes to tell. The “strong autonomy” versus “strong story” dilemma is a subject of on-going research (Mateas and Stern 2000).

Video games are an excellent platform to carry on research, as they combine rich multimedia with interactive experience. Furthermore, Artificial Intelligence (AI) algorithms can be run before and during the interactive experience with the purpose of making the experience tailored to

each individual player. In the last few years this has been a particularly active area of research (Riedl, Thue, and Bulitko 2011) and promising results have been already published.

One of them is the *Automated Story Director (ASD)* (Riedl et al. 2008)— an AI tool to help game designers ensure that the player can break the interactive narrative in conflict with the original story plot goals, in order to provide *agency*. It did so by automatically considering all possible breaks in the original story and proposing a single *contingency narrative* for each rupture. Such pre-processing was done at the narrative design phase, before the actual game begins. Thus, ASD was unable to tailor its contingency narratives to a specific player and relied on the one-narrative-fix-fits-all principle.

Around the same time, a system called *Player-Specific Stories via Automatically Generated Events (PaSSAGE)* (Thue et al. 2007) was proposed, which used AI techniques to model the player as he/she experiences a narrative-rich video game. Such a continuously updated *player model* was used to dynamically adapt the story, tailoring it to the current player. Unlike, ASD, PaSSAGE did not have any automation at the design stage and relied on a human designer to foresee all possible ways of a player breaking the story and manually design contingency narratives for each such break.

ASD and PaSSAGE are complementary and, in this paper, we propose, implement and demonstrate a principled way to combine them. At the narrative design phase (offline) we extend ASD with a library of *player archetypes*. For each possible rupture the archetypes are used to generate multiple diverse contingency narratives tailored for different types of players. Then, during the game time (online), we use PaSSAGE’s player model to select the most appropriate pre-computed narrative whenever the player breaks the exemplar narrative.

Problem Formulation

Our work considers a video-game like framework where a player is controlling a story character in an interactive environment. The sequence of player’s perceptions and actions over time constitutes an emergent narrative. The story designer intends to tell a story to create a certain experience for the player. The player is given some degree of autonomy – the power to change events of the story.

Current AAA video game productions can occupy over

a hundred developers for several years (Onyett 2007). Yet, in spite of such a massive creative and technical manpower, developers tend to reduce the complexity by taking away player's autonomy (e.g., by synchronizing several alternative stories via a "narrative funnel"). Such an approach can backfire as players tend to feel ownership of *their* story, and like agency manifested on a large scale (Orland 2012).

Thus, the first problem we tackle in this paper is to help the game developer anticipate story breaks, caused by player's interactions, and develop narrative responses to such ruptures.

For instance, consider the well-known story of *Little Red Riding Hood* – a folk story about a young little girl traveling through a forest to visit her grandmother. An interactive story of this narrative (Riedl et al. 2008) gives the player (playing as the Hunter) the autonomy to eliminate the wolf before it eats the grandmother. If during the game the player does choose to attack and eliminate the wolf, he/she would create a *rupture* in the exemplar narrative, breaking it. This can be fixed by switching to a *contingency narrative* that introduces a fairy to resurrect the deceased one, or supply another wolf to take its place. A *contingency narrative* is thus a complete, coherent story from start to finish created in response to a possible break in the exemplar narrative, caused by interaction with the story. When a contingency narrative is selected, the player is automatically "transferred" into the new narrative.

ASD generates such contingency narratives automatically, using knowledge of the virtual world in which the story takes place. An AI planner called *Longbow* is used for this process (Riedl et al. 2008), but it is limited on the diversity of content generation.

The second problem we tackle is tailoring the experience to each individual player. Current commercial games address this issue by augmenting the main story with a diverse array of side quests. For instance, in *Fallout: New Vegas* the player can elect to help the ghouls escape their cruel world for better places or catch a military spy leaking secrets via night-time radio transmissions. Such quests may appeal to different types of players and modern video games off-load the choice to the player themselves, expecting them to sift through a smorgasbord of quests and pick the most appealing ones by trial and error.

An alternative is to explicitly model the player and choose the most appropriate quests/story fragments for them behind the scenes (Thue et al. 2007). Continuing with our example, such an AI system may infer that a player is inclined to play as a fighter and thus replace the deceased wolf with a bigger and meaner wolf.

In summary, the problem we tackle in the paper is two-fold as we would like to help both the player and the developer. First, we would like to help game developers foresee possible ruptures and develop several contingency narratives suitable to varied player types. Second, we would like to improve players' experience by responding to each story rupture with a contingency narrative tailored to the specific player. We tackle both facets with AI methods.

Related Work

Drama managers (a system in charge of presenting a plot to a person) have been long-used as story-tellers (Laurel 1986). These systems have proven to be advantageous in some situations, as "drama management is beneficial when one desires certain properties or features of narrative experience to emerge." (Riedl 2010). Drama managers are a crucial part of interactive story-tellers, since they provide the guidance to the player and present the plot based on interaction. The need for content in such systems is evident, as they require a story database to select from during the online phase. *Experience managers*, on the other hand, are generalization of this concept; they look ahead for events that *could* happen and determine which ones *should* happen for a certain individual or conditions (Riedl, Thue, and Bulitko 2011).

Façade (Mateas and Stern 2003), was a drama manager based on three main ideas: breaking-up a narrative into "small pieces" or building blocks, adding an interactive setting, and engaging the audience into a compelling story. This seminal work required a tremendous engineering effort, prompting researchers to find more practical ways of building drama managers: a way of automatically generating content was in dire need.

One approach to automatic content generation is an analogy-based story generation of new narrative from a given *source* story and a *target* story, using similarities (analogies) and a common world knowledge (Ontanon and Zhu 2011). This approach lacks player model and it is not immediately clear how to apply this work in the context of drama management in video-game like settings.

Another recent experience manager is the *Prefix-Based Collaborative Filtering* (PBCF), which learns to present narrative based on a model created from user preferences and a database of previous ratings for a given narrative (Yu and Riedl 2006). This approach does not model the player during the game, guides the story plot, or generates contingency narratives automatically.

On the other hand, player agency and, most importantly, *perceived agency*, have been studied recently (Thue et al. 2010a; 2011), as it is central in any interactive story-telling environment. These investigations have concluded that a system that provides a player with the "sense" of making appropriate decisions (measuring appropriateness in terms of an accurate player model), creates the perception of greater control over the game events, making the experience more enjoyable to players. We deem this approach important and central to our work as well, reason why we take the same player modeling and event labeling for our current prototype, and expand it. Such models could also be later complemented with additional social-consistency (i.e. more believable characters) as has been proven by measuring fun and agency in user studies (Thue et al. 2010b).

Our Approach: PAST

The contribution of this paper is the first combination of two contemporary approaches to interactive story-telling. We use the player model of PaSSAGE to tell player-specific

stories, while using the planner of ASD to foresee player-induced story ruptures and create contingency narratives in response. The new system is called *Player-specific Automated Story-Telling: PAST*, and it is detailed in this section.

PAST operates in two phases. In the **offline** phase, it foresees possible ruptures and proposes fixes (i.e., contingency narratives) tailored to different player types. In the **online** phase, it models the player, detects player-caused ruptures and fixes them with the contingency narrative most suitable to the actual player.

Offline: Anticipating Ruptures

This module of PAST operates in the same way as the original ASD and we briefly review it here. The process begins with the story designer authoring an *exemplar narrative* or original story. To facilitate automatic detection of ruptures, the story is then represented as an AI plan (Riedl et al. 2008). This involves breaking the story down into *events* (e.g., “Red meets wolf”), *causal links* (e.g., “Hunter kills wolf” → “Wolf is dead”), *temporal links* and *authorial goals* (events that the author wants to happen). One example of such a plan is in Figure 1: events are represented by circles, causal links by solid arrows, temporal links by dashed arrows, and authorial goals by rectangles. The events are: Red greets Wolf (H), Red tells Wolf about Granny (T), Wolf eats Red (R), Wolf eats Granny (G), Hunter kills Wolf (K), Red escapes from Wolf (E), Granny escapes from Wolf (S) and Red gives a cake to Granny (C). The “islands” represent planning goals, e.g. Wolf eats Red (R) and Wolf eats Granny (G), whereas the outcome is the set of post-conditions as a result of all the events that took place.

The planner works by detecting story *intervals* in which there is a threatened condition. If a player violates such a threatened condition, a rupture occurs. For instance, for any interval with nodes H, T, R, G and K, the author required a wolf to be alive to meet the original goals; if the wolf is dead, it obviously cannot eat Red and Granny. The planner is aware that “killing the wolf” is a player action and the action *kill* will result in a dead character, anticipating a possible narrative rupture.

Offline: Generating Contingency Narratives

Having anticipated possible ruptures, PAST again refers to the planning representation of the exemplar narrative to generate possible fixes. This part is an extension of the corresponding module in ASD and operates as follows. For every anticipated rupture, PAST generates a contingency narrative. This is done by considering all possible ways in which a precondition required for the next event and violated by the player can be restored. If the player negates the precondition by killing the wolf, we can restore it by either spawning another wolf or resurrecting the deceased one via a fairy.

ASD arbitrarily selects a single contingency narrative among the possible fixes. PAST extends this approach by retaining several contingency narratives, each tailored to a different player *archetype*. To do so, PAST borrows the *player model* from PaSSAGE which was, in turn, based on the pen-and-paper RPG player types (Laws 2001). Each player is

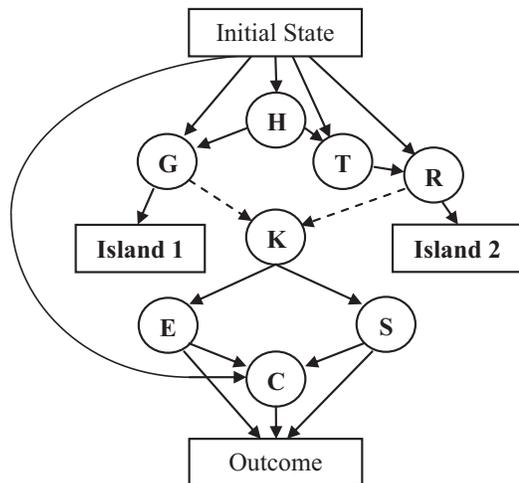


Figure 1: The exemplar narrative of the Little Red Riding Hood (Riedl et al. 2008).

represented as a vector of numbers in five dimensions. Vector components are numbers between 0 and 1, and indicate player’s inclination towards a certain style of play. The five styles are borrowed from PaSSAGE: *fighter*, *power gamer*, *story-teller*, *method actor* and *tactician*. For instance, the vector $(0.7, 0, 0, 0.5, 0)$ represents a person inclined to play as a fighter and in less amount, method-acting.

A continuum of player model vectors are possible making it impractical to generate a contingent narrative for each vector. Instead we define player archetypes represented by five orthonormal vectors in the space $[0, 1]^5$. The archetypes are: $A_1 = (1, 0, 0, 0, 0)$ representing a pure fighter, $A_2 = (0, 1, 0, 0, 0)$ representing a pure power gamer $A_3 = (0, 0, 1, 0, 0)$ for a pure story-teller, $A_4 = (0, 0, 0, 1, 0)$ for a pure method actor and $A_5 = (0, 0, 0, 0, 1)$ for a pure tactician. PAST generates and retains a single most appropriate contingency narrative for each of the five archetypes. In our running example, both possible contingency narratives — “Introduce another wolf” and “Introduce a resurrecting fairy” will be retained. The former suits the fighter archetype while the latter suits the story-teller.

To compute the suitability of a contingency narrative to a given archetype we use *annotations* on narrative events. Such annotations are also five-dimensional numeric vectors, each component indicating suitability of the event for different styles of play. These annotations have to be defined at design time by the author. In our example, the annotations of the events in the contingent narratives are as follows: $(1, 0, 0.8, 0.6, 0)$ for the event “Another wolf meets Red” and $(0, 0, 1, 0.1, 0.5)$ for the event “Fairy resurrects the wolf killed by the player”.

Given the annotations, a suitability of an event to a player type is defined as the dot product of the two vectors. For instance, the event ‘Another wolf meets Red’ has the suitability of 0.05 to the player modelled as $(0.7, 0, 0, 0.5, 0)$ since $(0, 0, 1, 0.1, 0.5) \cdot (0.7, 0, 0, 0.5, 0)^T = 0.05$. A narrative is comprised of events and so PAST computes the suitability

of a narrative for a player type as the average of suitabilities of the events comprising the narrative.

Using this notion of suitability at design time, PAST considers all anticipated ruptures and the corresponding contingency narratives. For each rupture, it computes the suitability of the relevant contingency narratives to each of the five archetypes. Then, the contingency narrative most suitable for each archetype is retained for later online use.

In our example, the rupture “Red kills wolf” can be fixed by two contingency narratives. One contains the event “Introduce another wolf” and the other contains the event “Introduce a resurrecting fairy”. PAST computes suitability of each of the two contingency narratives for each of the five archetypes and retains “Introduce another wolf” for fighters, power-gamers, and method-actors while keeping “Introduce a resurrecting fairy” for the other archetypes: story-tellers and tacticians.

In general, each of the resulting five contingency narratives can be broken again by the player. Thus, we run the planning process again, using each contingency narrative as the starting point. PAST expands such a tree of narratives until a certain limit on the tree height is reached. At this point, the entire tree is stored for online use. Figure 2 shows a part of such a contingency narrative tree. The exemplar narrative from Figure 1 is shown as the root of the tree. One of its ruptures (“Player kills the wolf”) can be fixed with five contingency narratives — one for each archetype: Fighter, Power gamer, etc. These are shown as tree nodes at the second level. Each of the contingency narratives can be broken again. Their contingency narratives are shown at level three.

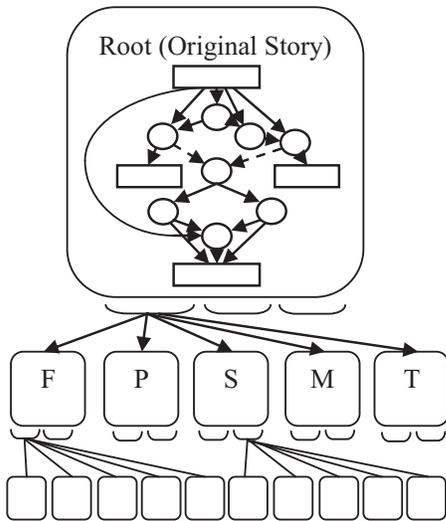


Figure 2: The tree of contingency narratives.

Online: Modeling the Player

In PaSSAGE, the player model is an annotation vector, adjusted each time a player takes an action. This is possible since player actions are annotated by the game designer with the five-dimensional vectors. For instance, action “Kill Wolf” can be annotated as $(1, 0, 0, 0.2, 0.6)$ indi-

cating a fighter inclination. Then, the model vector is updated by adjusting it towards the action annotation. The adjustment amount is a constant, fixed beforehand.

We adopt this approach and extend it in two ways. First, we use two adjustment amounts. The smaller amount is used for actions that do not break the current narrative. The larger adjustment amount is used when a rupture in the current narrative occurs.

Our second extension lies with increasing both adjustment amounts for shorter narratives. This is important as a shorter narrative contains fewer actions and events and thus a faster learning of player’s type is required in order for the model to be useful.

This process is shown in Figure 3 where the player causes a rupture by killing the wolf. At time t , the player is in the exemplar narrative. Their model is listed as P_t . The player kills the wolf, creating a rupture and triggering a model update. The new model vector is shown as P_{t+1} and is computed by adjusting the current model vector P_t towards the action annotation a , this is, $P_{t+1} = P_t + \alpha(a - P_t)$, where α is the learning speed.

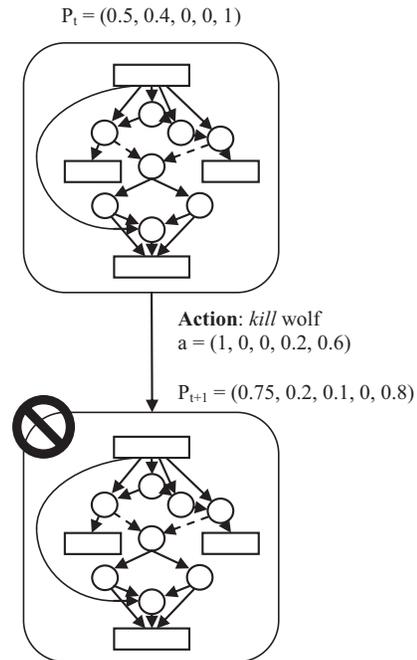


Figure 3: Upon killing the wolf, player’s model is updated and the exemplar narrative broken.

Online: Selecting Suitable Contingency Narrative

Online, once a rupture is detected, its pre-computed contingency narratives are retrieved from the tree of contingency narratives (e.g., Figure 2). PAST then uses its current model of the player to select the most suitable narrative. The suitability of each narrative is computed in the same way as at the offline stage. The selected narrative then becomes the current narrative, the player is transparently “transplanted”

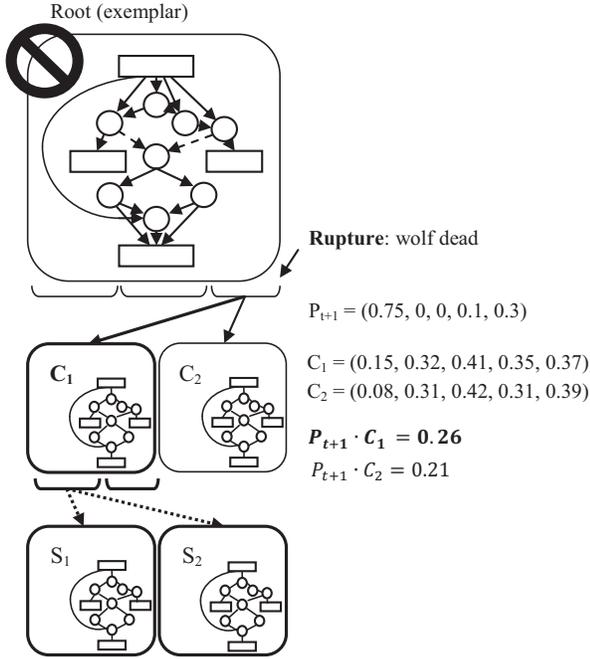


Figure 4: Rupture caused by the player and model update.

into it and the contingency unfolds from the appropriate (i.e., current) state.

However, this simple selection rule does not take into account the fact that the contingency narrative can itself be broken by the player, leading to follow-up narratives. To take these into account, we compute the suitability of a contingency narrative to a player as a discounted sum over itself and its follow-up contingency narratives. In the contingency narrative tree, these follow-up narratives are located in the subtree of the current contingency narrative. The discounting factor decays exponentially, rapidly reducing the influence of distant contingency narratives.

This process is illustrated in Figure 4 where the current narrative (from Figure 1) has just been broken by the player’s killing the wolf. The figure shows two of the five contingency narratives retrieved in response to this rupture. The naive selection method described above would select the narrative C_2 . However, the more informed selection method would also consider follow-up contingency narratives.¹ Given the discounted factor of 0.4, the influence of S_1 and S_2 will make C_1 appear more suitable for the player than C_2 . It will then become the active narrative.

Note that the contingency narrative selections made are player-specific. With a different player model (e.g., the player indicates a strong power-gamer inclination prior to killing the wolf) the suitability calculations will be different, leading to a possibly different contingency narrative.

¹For the sake of space, only follow-ups to C_1 are shown: S_1 and S_2 . There are follow-ups to C_2 as well.

A Preliminary Prototype

We implemented PAST on top of the ASD. The codebase includes the *Longbow* planner and is written in Lisp. We took the interactive Little Red Riding Hood story of ASD and extended it with several more actions and events, to allow for generation of more diverse contingency narratives.

To illustrate, in addition to a fairy that can resurrect the wolf, we also added the world knowledge that a fairy can *morph* into a big mean wolf: this is a practical example for demonstration purposes, but the actions could be whatever the author specifies them to be in the repertoire, as long as the action has the correct post-condition to re-enable the story for this potential rupture (i.e. a wolf being alive). The annotation of that event makes it more appealing to story-tellers/tacticians, and thus, gives PAST another choice among contingency narratives in response to “Player kills the wolf” action. This resurrecting fairy and the morphing fairy actions are shown in their Lisp format in Table 1.

Action: *morph*

```
(make-template :form '(morph ?fairy ?body)
:short '(?fairy morphs in a big mean ?body)
:after '(?fairy has morphed in a big mean ?body :period)
:during '(?fairy is morphing in a big mean ?body :period)
:annotation #(1 0 0.8 0.6 0))
```

Action: *resurrect*

```
(make-template :form '(resurrect ?fairy ?body)
:short '(?fairy resurrects ?body)
:after '(?fairy was resurrected by ?body :period)
:during '(?fairy is resurrecting ?body :period)
:annotation #(0 0 1 0.1 0.5))
```

Table 1: A sample text based events and annotations.

Conclusions and Future Work

We proposed a way of combining two existing approaches to interactive story-telling: ASD and PaSSAGE. In doing so, we aimed at helping narrative designers by automatically anticipating ruptures due to player actions, and generating a diverse set of repairs suitable to different player archetypes. We also aimed at increasing player’s enjoyment by selecting rupture repairs tailored to their play style. This was done by using the player modeling component of PaSSAGE. Our current prototype is fully operational, and demonstrates feasibility of this approach for merging ASD and PaSSAGE. We will be extending it with more content and subsequently evaluating its performance with a series of user studies. Our main goal is to demonstrate that PAST increases agency and fun while remaining practical for a game designer.

We hope our user studies will also allow us to fine-tune the model. One of the aspects that needs to be examined carefully is the effectiveness of *annotations*, since we use them to describe a narrative in terms of the average of smaller sub-events annotations. Simply put, this implies we might have annotations out of context (is “killing” always the same?); if

so, a method to account for this problem needs to be devised. The other element that needs to be tuned is the learning rate of the online learning function.

Acknowledgments

We are grateful for the funding by the National Research and Engineering Council (NSERC), consultations by David Thue, Mark Riedl and Matt Brown. Mark's sharing of the ASD code base is much appreciated. We also appreciate the feedback of the anonymous reviewers.

References

- Laurel, B. 1986. *Toward the Design of a Computer-Based Interactive Fantasy System*. PhD dissertation, Ohio State University.
- Laws, R. D. 2001. *Robin's Laws for Good Game Mastering*. Steve Jackson Games.
- Mateas, M., and Stern, A. 2000. Towards integrating plot and character for interactive drama. In *Working notes of the Social Intelligent Agents: The Human in the Loop Symposium*, 113–118. Menlo Park: AAAI Fall Symposium Series.
- Mateas, M., and Stern, A. 2003. Structuring content in the Façade interactive drama architecture. In *Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 93–98.
- Ontanon, S., and Zhu, J. 2011. The SAM algorithm for analogy-based story generation. In *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 67–72.
- Onyett, C. 2007. Blizzard's Worldwide Invitational – The StarCraft 2 Announcement. <http://pc.ign.com/articles/788/788627p1.html>.
- Orland, K. 2012. Protests over ending of Mass Effect 3 show fan investment in story control. <http://arstechnica.com/gaming/2012/03/protests-over-ending-of-mass-effect-3-show-fan-investment-in-story-control/>.
- Riedl, M. O.; Stern, A.; Dini, D.; and Alderman, J. 2008. Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on Systems Science and Applications* 4(2):23–42.
- Riedl, M.; Thue, D.; and Bulitko, V. 2011. Game AI as storytelling. In *Artificial Intelligence for Computer Games*. New York, New York: Springer USA. 125–150.
- Riedl, M. O. 2010. A comparison of interactive narrative system approaches using human improvisational actors. In *Proceedings of the Intelligent Narrative Technologies III Workshop*, 1–8.
- Thue, D.; Bulitko, V.; Spetch, M.; and Wasylshen, E. 2007. Interactive storytelling: A player modelling approach. In *Proceedings of the Third AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 43–48.
- Thue, D.; Bulitko, V.; Spetch, M.; and Romanuik, T. 2010a. Player agency and the relevance of decisions. In *International Transactions on Systems Science and Applications*, 210–215.
- Thue, D.; Bulitko, V.; Spetch, M.; and Webb, M. 2010b. Socially consistent characters in player-specific stories. In *Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 198–203.
- Thue, D.; Bulitko, V.; Spetch, M.; and Romanuik, T. 2011. A computational model of perceived agency in video games. In *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 91–96.
- Yu, H., and Riedl, M. O. 2006. Believable agents and intelligent story adaptation for interactive storytelling. In *Proceedings of the 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE06)*, 1–12. Springer.