Spring 2019

Measuring Interactive Narrative Quality with Experience Management as Story Graph Pruning

Jean-Paul Jeunesse
University of New Orleans

MEASURING INTERACTIVE NARRATIVE QUALITY
WITH EXPERIENCE MANAGEMENT AS STORY GRAPH PRUNING

An Honors Thesis

Presented to

the Department of Computer Science

of the University of New Orleans

In Partial Fulfillment

of the Requirements for the Degree of

Bachelor of Science, with University High Honors

and Honors in Computer Science

by

Jean-Paul Jeunesse

May 2019

Acknowledgment

I would first like to thank Dr. Stephen G. Ware, Alireza Shirvani, Edward T. Garcia, and Rachelyn Farrell for the great work they have done and are continuing to do, and for trusting me to contribute to that work however I can. Simply put, this paper would not exist without them and their hard work. I would also like to thank Dr. Ben Samuel for being my second reader and giving me important feedback during the writing process. Finally, I would like to, again, thank Dr. Ware for being my advisor. He was instrumental in this paper's development from its conception to its completion; I could not have done this without him.

Table of Contents

Abstract

An interactive narrative in a virtual environment is created through player and system interaction, often through an experience manager controlling the actions of all non-player characters (NPCs). Thus, the narrative (and its quality) is entirely dependent on a conflicting combination of unpredictability from the player and a controlled environment that must react to this unpredictability. Ideally, the experience manager should decide NPC actions in a way that never limits player freedom and shows the NPCs acting in believable manners to create a story that can be meaningfully affected by the player and feels organic. One solution to this is to view experience management as a story graph pruning problem. Nodes in the graph represent all the states that the virtual environment could possibly represent. These nodes are then connected by edges, which represent the actions that change one state to another. This graph is then intelligently pruned until NPCs have believable, unambiguous actions to take in every state, while never pruning player actions, with the intention of offering a more meaningful narrative.

Keywords: story graph, pruning, interactive narrative, experience management, game, virtual.

Introduction

The narrative in a virtual environment is unique from other art forms or media in that it can be, and often is, interactive. The player generally can control a virtual character that can actively affect the narrative through their actions. An interactive narrative can be used strictly for entertainment, or for more serious purposes like training and education. In the game used for this study, the player controls one character, while the game's experience manager controls the non-player characters (NPCs) in accordance with their beliefs and goals defined by the systems they inhabit. Together, through the actions the player chooses to make and the decisions of the experience manager, a narrative is created. This is to accomplish the ultimate goal of an interactive narrative, that is as Riedl and Bulitko (2012) state, to "immerse the user in a virtual world such that he or she believes that they are an integral part of an unfolding story and that their actions have *meaningful* consequences" (p. 1). However, while the player chooses their actions freely, how does the experience manager decide what actions the NPCs carry out to best service the story and achieve this goal?

This paper approaches this problem by first viewing experience management as a story graph traversal problem. Nodes in the graph represent the various states that the virtual environment, NPCs, and player can be in. These nodes are then connected by edges, which represent the actions that change one state to another. Actions can be taken by the player, NPCs, or both. Thus, in every state, the player can execute an action, the system can execute an NPC action, or they can together execute a joint action. Through player and system interaction, a path is formed that leads to a terminal state (the end of the game).

I will be comparing two experience managers represented as navigating two different story graphs to determine which yields higher quality narratives. Both story graphs are fully generated and then pruned in advance, rather than during runtime, to account for every path the player can possibly take in the game. Pruning is the act of removing edges (as well as states that can no longer be reached as a result of said removed edges) from the graph. Both graphs' player edges were never pruned, so as not to limit the player's actions. The first story graph is pruned only once based on NPCs' intentionality (see the Intelligent Story Graph Pruning section for more detail), while the second is also pruned this way, then further pruned using various other techniques until each NPC in each state can perform at most one action that is also believable for that character. The graphs I will be using are from ongoing work done by Ware, Shirvani, Garcia, and Farrell, which I will be covering in more detail throughout the paper, mainly describing the pruning techniques applied to the graphs.

The purpose of this paper is to contribute quantitative results to the work of Ware et al., testing if their pruning can potentially lead to higher quality narratives. I also describe the criteria used to determine the narrative's quality, and why I use these criteria. To test the quality of the narratives, I will run multiple simulations of the player taking randomly selected actions in the game using both story graphs. I will then take the average quality of all the simulated games from both graphs and compare the results.

Literature Review

Story and plot graphs are common data structures used for managing interactive narratives. Throughout the literature though, these terms are often used inconsistently with both taking on varying definitions. The work done by Ware et al. (and by extension, this paper) use the term story graph as defined by the work of Riedl and Young (2006). They describe it as a branching story structure, a directed graph of nodes that represent the various states of the virtual world, connected by edges/arcs, representing any character actions that may change the current state.

Bates (1992) and Weyhrauch (1997) were some of the first to use graph traversal to represent experience management. Bates framed experience management like a game of chess between the player and the game "director" (the computational model of drama controlling the elements of the virtual world); they take turns performing actions in response to one another to create an interactive narrative. Bates's Oz Project, which Weyhrauch also worked on, involved using a plot graph model to navigate the story. The plot graph contained only the major scenes of the story, partially linked together as a directed graph. Weyhrauch continued this research with his experience manager Moe, which used an abstract adversary search that evaluated an experience's quality to determine the ideal path for a user.

Nelson et al. (2006), Roberts et al. (2007), and Thue and Bulitko (2012) built on this work using Markov decision processes (MDPs). The work of Nelson et al. used declarative optimization-based drama management (DODM) to guide players through a narrative. DODM abstracts a story into specific plot points, evaluates the quality of potential plot-point sequences, and then re-structures the world to encourage the player to pursue the highest quality sequence. The work of Roberts et al. navigated a story through targeted trajectory distribution MDPs, specifically aiming for a variety of experiences to support subsequent playthroughs. The work of Thue and Bulitko focused more on changing the game's dynamics (how the player's actions affect the game world) to better suit the player. Their Procedural Game Adaptation framework gathers information about the player as they play and then uses that information to inform how to change the game's dynamics in the future (e.g. if in one scene the player chooses fighting over negotiating, they may encounter more events in the future that require combat).

Arinbjarnar, Barber, and Kudenko (2009) also survey and compare a number of systems based on experience management as graph traversal. Many of the ones surveyed are described as a plot graph structure, but each structure has varying definitions for how the graph is represented. However, they do all view graph traversal as a combination of player and system interaction.

Another system commonly used is dynamic narrative generation; Kybartas and Bidarra (2017) survey a number of these. These systems generate their narratives at run time, unlike the story graphs used in this study which were fully generated and pruned in advance. Even though these dynamic experience managers do not initially generate a full story graph, they are still navigating one, albeit one being generated on demand. This avoids the cost of calculating an entire graph in advance, which may not even be an option for most games' larger story domains. However, having access to the full story graph allows one to consider the long-term consequences of an action.

Dynamic experience managers can employ two similar techniques for dealing with large story graphs and unpredictable players: reactive and proactive mediation. Reactive mediation involves first forming a plan for the narrative based on what the player is expected to do; when the player deviates from that plan, the system will either accommodate the player or intervene (Riedl, Saretto, and Young 2003). The system accommodates the player by re-planning the story according to the player's unplanned action. However, if accommodation is not possible, the system must intervene and prevent the player from even doing the action that disrupts the story. Consider a situation where the player is about to shoot a character that must later perform an action that is important to the story. Accommodation would allow the player to do so, then maybe have another character perform the important action later in the story. Intervention would prevent the player from ever shooting the character, maybe through the player always missing their shot or the gun jamming, not allowing the player to shoot.

Ideally, the system will always try to accommodate the player, as intervention subverts the player's mental model of the environment's rules and decreases agency, the player's feeling that they can take meaningful action to affect the trajectory of the story (Wardrip-Fruin et al. 2009). When traversing a story graph, intervention would be the act of pruning a player action edge. Proactive mediation attempts to reduce intervention as much as possible. While reactive mediation only responds to player actions as they happen, reacting to actions that may disrupt the story at the last possible moment, proactive mediation re-structures the story by anticipating actions that the player may take to better accommodate the player before they get to a point where intervention is necessary (Harris and Young 2009). The pruned story graph used in this study is a type of proactive mediation. The system can always accommodate the player since the entire story graph is considered and never prunes player edges.

Story Domain

A story domain defines objects and actions used to construct, in this case, a story graph. The objects consist of characters, places, things, and concepts (such as a character's beliefs), while the actions are those that can be taken by the characters. Characters are intelligent agents that have beliefs and goals. A state defines the current configuration of the world. For this domain, that includes information about if a character is alive, armed, or a criminal, where the characters and items are located, and the characters' beliefs about the location of items and characters as well as the criminality of other characters.

The domain used for the story graphs in this study is inspired by a certain subset of characters from Ware and Young's (2014) interactive narrative *The Best Laid Plans*, in which the player character's grandmother is sick, thus the player is tasked with purchasing medicine and successfully bringing it to her. The game includes three NPCs. The first is a merchant selling medicine and a sword for one coin each at the market. The second is a town guard, also at the market, watching for criminals. The third is a bandit, who will try to steal coins or medicine from other characters. When he is not trying to steal, he waits at the camp by his chest, which contains one coin. A crossroads connects the market, camp, and the player's house. The game ends when the player dies or successfully brings the medicine home. Despite the seemingly small domain size, there are many interesting paths to take in order to arrive at either of these endings.

Intelligent Story Graph Pruning

This section will cover the work done by Ware et al. relevant to this paper, defining certain terms and their design goals for the game, as well as discussing the techniques used to prune the story graph.

The story graph consists of nodes and directed edges leading to other nodes. Nodes represent the states that define the configuration of the virtual world and every character's beliefs about that configuration. These characters can perform actions that define when the action can happen, what changes in the world when it happens, who is performing the action, and who observes the changes that occur. Take, for example, the action where the player kills the merchant at the market and the guard sees it. The player, merchant, and guard must be at the market, the player must be armed, and the merchant can be armed or unarmed for this to happen. The configuration of the state also changes in multiple ways as a result of this action: the merchant dies, the player becomes a criminal, and the guard now believes the player to be a criminal, since he was an observing character.

The edges in the graph represent the actions, moving from one node to another, thus changing one state to another. A player edge is an edge that represents one player action, while an NPC edge represents one NPC action. A mixed edge is both a player and NPC edge that involves actions from the player and NPC; for example, the player buying something from the merchant.

A full story graph contains every possible state and every allowable edge. The goal was to generate a full story graph, then prune NPC edges until the experience manager has unambiguous directions for what each NPC should do in each state. The various techniques in which the graph was pruned were done in service to three design goals of the game: the game must always be finishable, NPCs should act believably, and the player, not the experience manager, should be responsible for how the story unfolds. Since a player's actions should never be limited, edges that only involved a player action should never be pruned. The NPC edges were pruned using nine different techniques in the order presented below.

First, NPC edges that did not align with an NPC's intentionality were pruned. An NPC's intentionality is its tendency to work towards its goals, an important property of believable character behavior (Riedl and Young 2010; Ware and Young 2016). For example, one of the guard's goals is to attack criminals. If the guard believes that the player is not a criminal but attacks them anyway, that would be unintentional behavior and would be pruned. Likewise, if the guard believes the bandit to be a criminal, knows his location, and then attacks him, that would be intentional behavior and would not be pruned.

In practice, the full story graph for this domain was too large to generate, so the initial graph was generated using intentionality pruning. Thus, the initial graph is a story graph that would result from pruning the full story graph based on intentionality. As such, when I refer to the full story graph, it will refer to the one pruned just based on intentionality. The graph that is further pruned using the following eight techniques will be referred to as the pruned story graph. These two graphs are what will ultimately be compared based on narrative quality.

The second pruning technique was called shorter plan pruning. If an NPC can accomplish the same goal by means of two different plans, the longer plan is pruned for efficiency. For example, if the guard is in the crossroads and wants to return to his post at the market, he can walk straight to the market in one action. However, he can instead walk to the camp, walk back to the crossroads, then walk to the market, taking three actions. Both achieve the same goal, so the longer plan gets pruned.

The third pruning technique was based on the “Lazy NPC principle.” If a character has two plans that achieve a goal, the plan that involves fewer player actions is pruned in order to give the player more options and ways to explore. For example, if the guard wants to know the location of the bandit and the player knows his location, either the guard can follow the player around waiting for them to tell him, or the guard can stay at his post and wait for the player to come to him. Both plans exhibit intentional behavior and accomplish a goal, but the former lessens the actions the player can take to achieve their goal and is thus pruned.

The fourth and fifth pruning techniques were tie-breaking prunes, meaning they will never prune the last edge for an NPC. Given two NPC actions, the fourth technique (called unique endings pruning) prunes the action that most decreases the number of unique endings, of which there are two: either the player dies, or they bring the medicine home. For example, consider a state where the guard believes both the player and bandit are criminals and has actions available to attack either one. Both of these actions achieve a goal, but attacking the player reduces the number of unique endings to one (the player dies). Thus, that action is pruned in favor of attacking the bandit, which allows two unique endings to still be available. This aligns with the design principle that the player should be responsible for the ending attained, not an NPC.

The other tie-breaking technique was player model pruning. Similarly, given two NPC actions, the action which reduces the number of unique models that can apply to the player is pruned. While player modelling has been used to dramatically shape narratives to each specific player, like with the interactive storytelling system PaSSAGE created by Thue et al. (2007), this work was not intended to contribute to that research. It was used only to show that player modelling can be applied to story graph pruning. As such, the player modeling in this game is very simple: the only model that is applied to the player is whether or not they are a criminal. Thus, this pruning tries to remove NPC actions that force a player into one of these models: either having to commit a crime to finish the game, or no longer being able to commit a crime. An example would be a situation where the bandit has killed the guard and robbed the player of their coin, then the merchant kills the bandit and takes both of the coins. The player can now only get the medicine by looting either the guard or bandit to get a sword and attacking the merchant for the medicine, making the player a criminal. It should be noted though, that while player model pruning was applied to the graph, it did not actually result in any edges getting pruned.

The sixth technique was goal priority pruning. If a character has more than one goal, they are ranked from most to least important. Given two edges that both accomplish a goal, the goal with lower priority is pruned. For example, the bandit can either protect his chest at the camp or try to rob other characters of their coins or medicine. In a situation where he has left his camp to rob the player, the experience manager can choose two different actions that align with his goals:

to either rob the player or walk back to his camp. If the experience manager chooses the latter, a cycle will begin where the bandit continues to leave his camp briefly only to come back soon after. As the latter is his lower priority goal, it is pruned, preventing this behavior. However, goal priority pruning does not prune all cases of these cycles. Thus, the seventh technique (cycle pruning) aims to prune the rest of NPC edges that exhibit this behavior. A cycle that repeats for 3 or more edges gets pruned.

Finally, two last prunes were applied: arbitrary and dead-end pruning. If after all the prior pruning, an NPC still has more than one action to take in a state, they are considered to be equally reasonable and believable, so one is chosen arbitrarily, as the rest are pruned. Also, any NPC actions that could lead to “dead-ends” (a node from which it is impossible to reach an ending) are pruned.

Evaluation

The goal of this paper is to compare the effectiveness of using the pruned story graph and the full story graph, specifically, in terms of their narrative quality and how often each graph has to intervene in order to ensure that the game can still be finished. My claim is that the pruned graph will produce higher quality narratives than the full graph. I will also be verifying that the pruned graph will never have to intervene, since all dead-end states were pruned; the game can always be completed regardless of what actions the player chooses to take. Whereas I posit that the full graph will have to intervene, however minimally, to avoid reaching a state that cannot reach an ending.

Simulating Games

In order to compare these two graphs, I simulated games in which the player would perform a random action in each state, as a player’s actions are inherently unpredictable as far as the experience manager is concerned. For the full graph, NPCs may have more than one action they can take in any state, and since the full graph was only pruned based on intentionality, these actions are regarded as equally intentional. Thus, one can be chosen at random to be performed, like the player's actions. The NPCs operate in the same way in the pruned graph but have only one potential action to perform or none, since the graph was pruned to have at most one action per NPC per state.

The objective of each simulation was to determine both the average quality of the game and whether the game achieved an ending or reached a dead-end state that made the game unable to be finished. The significance of the latter is that if a game gets to a dead-end state, intervention would have had to happen (i.e. a player edge would have had to be pruned) to avoid getting to that state. If it does reach an ending though, no intervention would have had to occur.

I determined the quality of a game based on two criteria: if both, either, or neither of the two endings are possible and if both or either of the two player models are available to the player for every state visited, averaged over the course of a game. So, each state's quality is the sum of the number of unique endings available (0, 1, or 2) and the number of unique player models available (1 or 2, as one model, the player's current model, is always available to the player). Therefore, the quality of a state has a range of 1 to 4, 1 being a dead-end state with only one player model available, and 4 being a state that can still reach both endings and has both player models available. These criteria were chosen as they align with the design goals of the game. It is important that the player be in control of how the story progresses. Thus, a state that has more options available to the player (which, for this testing, is represented as which endings and player models are still possible) is considered to be of higher quality.

For this study, I had access to both graphs' text files, providing me with relevant information like all the states' values and the next states that they can reach. Thus, I simulated these games by traversing both graphs directly, reading from these files. I wrote a program that could navigate both story graphs and would return a game's average state quality and whether or not it reached an end state or dead-end state. In order to use this program though, I had to first determine all the states' qualities and all the end and dead-end states.

For both graphs, I needed two different lists of all the possible endings marked as either positive or negative. The lists differed in how I determined the endings' signs. For one list, I marked any ending as positive if it was the one where the player successfully brings the medicine home or as negative if it was the one where the player dies. For the second list, the endings were marked based on the player model: positive if the game ended with the player not being a criminal, and negative if it ended with the player as a criminal.

Next, for both graphs, I had to determine the quality of each state, initially only based on how many types of endings were possible. For each positive ending from its respective list, I assigned it a quality of 1, as only one ending is possible in an end state. I then assigned all the states that could reach each positive ending a quality of 1 too, as they can reach at least this ending. This resulted in all possible states being assigned either a quality of 1 (meaning the state can reach a positive ending) or 0 (meaning the state cannot reach a positive ending). I then did the same for all the negative endings and combined the results. Now each state has a quality of 0, 1, or 2 based on how many types of endings it can reach. This is also how I determined which states were dead-ends, since any state with a quality of 0 cannot reach either type of ending.

I then determined the quality of each state based on how many unique player models were possible in the same manner, albeit using the other list of positive and negative endings.

The only difference was that, for the full graph, I had to take into account the dead-end states. It may be possible that the player model could still change in a dead-end state. Thus, I made a list of all the possible dead-end states and also marked them as positive or negative based on the player model in each one. They were then assigned a quality of 1 or 0 in the same way as the endings and the states that connect to them. After combining the results of the positive and negative endings and dead-ends, each state has a quality of 1 or 2 based on how many unique player models are available.

Finally, for both graphs, I calculated the sum of the number of unique endings available and the number of unique player models available in each state to get the overall quality of every state. With this information, my main program can run its simulated games. It begins a game at the initial state (which, when the game begins, is state 0) and increments the amount of states visited. It then gets the state's overall quality and adds it to the sum of all the states' qualities in this game. Next it checks if that state is an ending or dead-end (since the pruned graph has no dead-ends, it only checks if the state is an ending when using that graph). If it is the former, the game has arrived at an ending, meaning the experience manager never would have had to intervene. If it is the latter, the game cannot end, meaning the experience manager would have had to intervene to prevent the player from getting to this state. If it is neither, a new state is randomly chosen from the possible states that the initial state can reach (representing a random action done by the player). The process is then repeated for this new state until an ending or dead-end is reached.

At the end of a game, the program also calculates the average quality of that game by dividing the sum of all the states' qualities by the number of states visited. After running any number of games, it returns the average quality of all the games, how many games reached an ending, and how many did not reach an ending.

Results

I ran a total of 1000 simulated games for each story graph. The pruned story graph had an average quality of 3.485 across all games, reached an ending for all 1000 games, and never reached a dead-end. The full story graph had an average quality of 3.617 across all games, reached an ending for 993 games, and reached a dead-end for 7 games. The full story graph had a higher average quality by a difference of 0.132. The pruned story graph arrived at an ending in 100% of the games, while the full story graph arrived at an ending in 99.3% of the games.

Conclusion

These tests show that while the full story graph intervened minimally (less than 1%), there is a possibility of it happening. On the other hand, the pruned story graph guarantees that it will always accommodate and never intervene. The full story graph acts as an existential quantification for possible paths the player can take: there exists a path that can lead to an end state, but not every path leads to one. The pruned story graph acts as a universal quantification for possible paths: for all paths, an ending can be reached.

These tests also show that the full graph, on average, leads to higher quality narratives. Since quality is relative to how it is defined in this paper, that definition's limitations may have caused this result. For example, if the player continuously goes back and forth between high quality states, that game would technically have a high-quality narrative, but that behavior should not be allowed to artificially inflate the quality of a narrative. As such, I simulated more games trying to address this limitation by including a state's quality only once in the game's average, excluding subsequent visits to the state. However, the results were largely the same, albeit slightly higher: the pruned graph had an average quality of 3.536 and the full graph had an average quality of 3.634.

While this limitation did not appear to affect the results, others may have. The way I am defining quality could lead to a bias for longer narratives in which the player does not make impactful decisions that could permanently affect their player model or shut off the possibility of an ending; the longer they take to make those decisions, the higher the quality could potentially be. Future work could benefit from addressing any biases or limitations by testing differently and defining quality in a new way, maybe by not having quality be defined for every state and then averaged, but by having it increase or decrease based on certain actions taken.

However, assuming this measure of quality is adequate, these results could show that the pruning techniques applied after intentionality pruning negatively affect the quality of the narrative; at best, they do not make a significant impact on it. If this is the case, the only pruning techniques that potentially benefit the narrative's quality would be intentionality pruning and dead-end pruning, since the latter is responsible for the pruned graph never having to intervene. If future work continued to use the same measure of quality, it could compare multiple graphs that were all pruned based on intentionality first and dead-ends last. Each would also have only one of the remaining pruning techniques applied to it to compare the techniques in isolation, determining if any have a positive or negative effect on the narrative's quality.

Overall, these results do confirm that the full graph may have to intervene while the pruned graph never will. They also show that the full graph ultimately produces higher quality narratives than the pruned graph. However, there are potentially other ways of defining quality quantitatively that could lead to new insights on how pruning a story graph affects said quality.

References

Arinbjarnar, Maria, Heather Barber, and Daniel Kudenko. "A critical review of interactive drama systems." *The Society for the Study of Artificial Intelligence and Simulation for Behavior '09 Symposium: AI & Games, Edinburgh, UK*. 2009.

Bates, Joseph. "Virtual reality, art, and entertainment." *Presence: Teleoperators & Virtual Environments* 1.1 (1992): 133-138.

Harris, Justin, and R. Michael Young. "Proactive mediation in plan-based narrative environments." *IEEE Transactions on Computational Intelligence and AI in Games* 1.3 (2009): 233-244.

Kybartas, Ben, and Rafael Bidarra. "A survey on story generation techniques for authoring computational narratives." *IEEE Transactions on Computational Intelligence and AI in Games* 9.3 (2017): 239-253.

Nelson, Mark J., et al. "Declarative optimization-based drama management in interactive fiction." *IEEE Computer Graphics and Applications* 26.3 (2006): 32-41.

Riedl, Mark, Cesare J. Saretto, and R. Michael Young. "Managing interaction between users and agents in a multi-agent storytelling environment." *Proceedings of the second International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, (2003): 741-748.

Riedl, Mark O., and Robert Michael Young. "From linear story generation to branching story graphs." *IEEE Computer Graphics and Applications* 26.3 (2006): 23-31.

Riedl, Mark O., and Robert Michael Young. "Narrative planning: Balancing plot and character." *Journal of Artificial Intelligence Research*, 39 (2010): 217-268.

Riedl, Mark Owen, and Vadim Bulitko. "Interactive narrative: An intelligent systems approach." *AI Magazine* 34.1 (2012): 67.

Roberts, David L., et al. "Authorial idioms for target distributions in TTD-MDPs." *Proceedings of the National Conference on Artificial Intelligence*. (2007): 852-857.

Thue, David, and Vadim Bulitko. "Procedural game adaptation: Framing experience management as changing an mdp." *Eighth AAAI Artificial Intelligence and Interactive Digital Entertainment Conference*. (2012): 44-50.

Thue, David, et al. "Interactive Storytelling: A Player Modelling Approach." *Artificial Intelligence and Interactive Digital Entertainment*. (2007): 43-48.

Wardrip-Fruin, Noah, et al. "Agency Reconsidered." *Digital Games Research Association Conference*. 2009.

Ware, Stephen, et al. “The Best Laid Plans.” 2014. https://nil.cs.uno.edu/projects/blp/

Ware, Stephen G., and R. Michael Young. "Intentionality and conflict in The Best Laid Plans interactive narrative virtual environment." *IEEE Transactions on Computational Intelligence and AI in Games* 8.4 (2016): 402-411.

Weyhrauch, Peter William. *Guiding interactive drama*. Ph.D. Thesis at Carnegie Mellon University, 1997.