Multiagent Narrative Experience Management as Story Graph Pruning

Stephen Ware, *Member, IEEE*, Edward T. Garcia, Mira Fisher, Alireza Shirvani, and Rachelyn Farrell

***Abstract*—In interactive narratives, experience management is used to control the world and the nonplayer characters (NPCs) a player interacts with, encouraging particular types of stories or discouraging others. The space of all stories in a narrative can be understood as a story graph, with story states as nodes and actions in the story as directed edges. In this article, we present experience management as a graph pruning problem. Starting with the full story graph, edges representing NPC actions may be pruned until there is at most one action per NPC per state. With the full graph available, the choice of what to prune may consider all possible futures, and we can ensure that undesirable stories are not reachable. By never pruning player actions, we ensure the player may make any choice and still be accommodated in the story. When this method was used to manage the story of an adventure game, players found our technique generally produced higher agency and more-believable NPC behaviors than a control. Finally, we discuss scaling the results of this method for practical use.**

***Index Terms*—Experience management, graph pruning, strong story mediation.**

I. Introduction

INTERACTIVE narratives are a core component of many games and other virtual environments, where they play a central role in meeting goals for entertainment, education, and therapy. In these environments, a player takes on the role of a character in the narrative, while the system controls all other nonplayer characters (NPCs) and the environment itself. The system must act as an *experience manager*, directing the story to ensure that the experience the player has within the narrative fulfills the design goals of the game or tool. The techniques used for this management can be compared against two fundamental styles: 1) strong autonomy approaches; and 2) strong story approaches [1].

Strong autonomy systems focus on providing a strong cast of believable characters and empowering these NPCs and the player to drive the story forward freely. Without strict guidance and hand-authoring, the emergent narrative that arises may take many unique and unexpected forms. By providing a realistic, or at least a *convincing* simulation of behavior and interactions with relevant characters, each of these unique experiences may have value in achieving the designers' goals without direct intervention. Games like *Dwarf Fortress* [2] and the scenario simulation component of *Tactical Action Officer* [3] fall into this category.

Strong story systems aim to meet strict constraints on the structure and content of their stories. By planning in advance around each possible branch in the narrative, designers attempt to ensure that the resulting story experience will meet their requirements regardless of what action the player takes. This planning is often done at the time of design. Every player choice must be anticipated in advance and each response preauthored, preventing the story from going in an undesirable direction. Interactive narrative games like *The Walking Dead* [4] fall into this category. Hand-authoring can be a brittle method, requiring exponentially more effort for each additional branch of the story [5], and can lead to out-of-sequence stories when players behave unexpectedly [6]. This method also limits the player to the designer's imagined paths, and the authoring effort limits the breadth of those imagined paths.

In the last few decades, there has been increased interest in automated strong story systems that use artificial intelligence experience managers to ensure, during play, that the story meets the designer's constraints. Games like *Façade* [7] and *The Best Laid Plans* [8] use this approach. By employing computational models of narrative to reason about the story as it unfolds, they anticipate possible futures, and can use this to ensure narrative structure and encourage or avoid specific interactions. Some systems, like *Façade's* drama manager, only anticipate the immediate future, and authors must account for further effects. Planning systems like *The Best Laid Plans* anticipate sequences of events and reason about their effects, looking further into the future. Planning enables the techniques in this article, which presents a primarily strong story system, augmented by consideration for NPC autonomy which is achieved through making each character a planning agent.

In this article, we cast the process of progressing through an interactive narrative as a graph traversal problem for two agents—the player and the experience manager—applied to a story graph. A story graph is a directed graph where distinct states of the story world are represented by nodes, and these nodes are connected by edges indicating the actions that

Manuscript received 7 October 2021; revised 3 March 2022; accepted 12 May 2022. Date of publication 8 June 2022; date of current version 15 September 2023. This work was supported by the NSF under Grant IIS-1464127, Grant IIS-1647427, Grant IIS-1454127, Grant IIS-1647427, and Grant IIS-1911053. *(Corresponding author: Mira Fisher.)*

Stephen Ware, Mira Fisher, Alireza Shirvani, and Rachelyn Farrell are with the Narrative Intelligence Lab, Department of Computer Science, University of Kentucky, Lexington, KY 40506 USA (e-mail: sgware@cs.uky.edu; ocfi222@g.uky.edu; ashirvani@cs.uky.edu; rachelyn.farrell@uky.edu).

Edward T. Garcia is with the University of New Orleans, New Orleans, LA 70148 USA (e-mail: etgarci1@uno.edu).

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the University of New Orleans Institutional Review Board, Application No. 1Dec18.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TG.2022.3177125.

Digital Object Identifier 10.1109/TG.2022.3177125

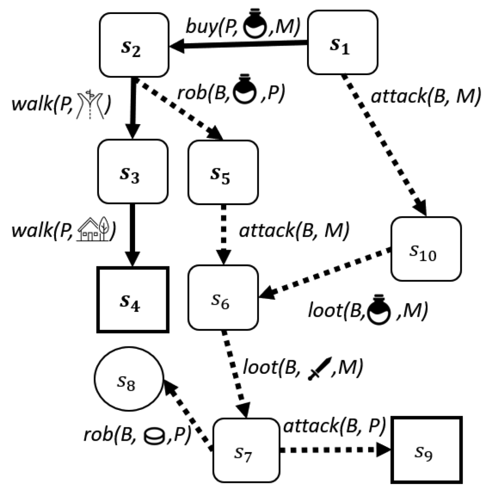


Fig. 1. A snippet example of story graph structure, with the path of a specific narrative in solid lines, and all other edges dashed. Character names are abbreviated, and symbols stand in for the potion, sword, coins, crossroads, and cottage. Square nodes are terminal, circular nodes have no path to a terminal node.

transition the world from one state to another [9]. Player actions, and NPC actions selected by the experience manager, determine a path through the story graph until they reach a terminal node. Terminal nodes represent the endpoints of the narrative—situations in the story where play ends, for reasons such as the death of the player character or another conclusion of the story. As highlighted in Fig. 1, the story graph shows many possible stories, and any one path through the graph indicates one of these possibilities. This framing allows us to cast experience management as a story graph pruning problem.

By removing NPC edges from the story graph until each NPC has at most one action to take in any state, we can define an action selection policy for the experience manager. By never removing player edges, we ensure that the player is always free to take any possible action in any state, and all emergent interactions may be anticipated. The method also makes the presence of dead end states more apparent, such as would be indicated by the node s_8 in Fig. 1. At this point in the snippet, the story domain does not allow for an ending to be reached. However, by pruning the bandit NPC's rob action leading to it, this exceptional state can be avoided.

Story graph pruning is difficult. Even for small interactive narratives the associated story graph can be intractably large, and the criteria for when an action should be pruned may be based on all the remaining potential paths of the story, as well as the current and prior states. In this article, we begin by exploring a scenario where the entire story graph can be generated and kept in memory. By keeping the entire graph in memory during pruning, we can consider all the long-term consequences that removing an edge will have on the space of possible stories. We present various pruning criteria and the motivations behind them. We then give the results of a study which show that players find NPC behavior more realistic and experience higher agency when graphs are pruned using our methods. Finally, we examine how the insights gained from this study can be applied to online systems where generating the graph in advance is intractable, and discuss how a generated graph may be utilized in an online system.

A. Prior Publication

This article builds on a previously published conference paper [10]. This work has been further developed with a discussion of *epistemic* states and their relationship to the story graph, the details of the *player model* pruning step, and the results of the analysis of the prevalence of *dead end* states in the story graph. Finally, we include new methods to support the practical use of this system through machine learning.

II. Related Work

A. Story Graph-Based Systems

Terms like *story graph* and *plot graph* have been used inconsistently in the literature [11]. We adopt a modified version of Riedl and Young's [9] definition of a story graph. A story graph is directed graph whose nodes are world states and whose edges are actions. An action edge $s_1 \xrightarrow{a} s_2$ may connect state s_1 to state s_2 if action a is possible to take in state s_1, and taking action a in state s_1 would change the world state to s_2. Story graphs are a common data structure for representing interactive narratives, including nondigital ones like Choose Your Own Adventure books [12].

Bates [13] and Weyhrauch [14] were some of the first to describe experience management as a graph traversal problem jointly solved by the player and an AI experience manager, though their *plot graphs* were defined differently than our story graphs. Weyhrauch used search-based optimization to find ideal paths through plot graphs, while Nelson *et al.* [15], Roberts *et al.* [16], and Thue and Bulitko [17] used MDP-based methods to determine graph traversal policies. Arinbjarnar *et al.* [18] survey systems based on graph traversal. While they differ in their graphs' structures, all frame the solution as a joint decision-making problem between the player and the system.

Some systems require story graphs to be acyclic, implying some constraints on history, but we do not. Like many previous systems, our system relies on the Markov assumption—the experience manager makes decisions based only on the current state and its descendants and does not track the history of how the player arrived at that state. The Markov assumption is a limitation for story graph systems. Farrell *et al.* [19] demonstrated that different actions leading to the same state can suggest very different futures for a story. However, given that our graphs are hundreds of millions of nodes and they already strain the limits of what can be feasibly computed, we accept this simplifying assumption for this initial work.

B. Mediation-Based Systems

Systems that do not explicitly use story graphs may still use them implicitly. In particular, generating a narrative at run time can be understood as navigating a story graph which is generated on-demand. These dynamic systems avoid the potentially prohibitive cost of calculating the entire graph in advance, but may find it more difficult to reason about the long-term consequences of an action. Kybartas and Bidarra [20] survey dynamic narrative generation systems.

To manage large story graphs and unpredictable players, many experience managers form a plan for the narrative based on what they expect the player to do and employ *reactive mediation* when the player deviates from that plan [21]. Ideally, the system *accommodates* the player by replanning the story to include the unexpected action (e.g., an important NPC is killed, so another NPC takes their place in required events). When this is impossible, the system may *intervene* to make the player's action fail (e.g., a gun jams and fails to fire). Intervention subverts the player's mental model of the environment's rules and may harm their perception of *agency*, the player's feeling that they can take meaningful action to affect the story [6]. In a graph traversal context, intervention can be viewed as pruning a player action edge from the story graph—that action should have been possible and should have transitioned the story to a new state, but the system removed it to prevent the narrative from being irrecoverably disrupted.

Experience managers can also employ *proactive mediation* [22]. By fully anticipating the player, the system can avoid intervention by ensuring that every player action can always be accommodated. The story graph pruning techniques we describe in this article are a kind of proactive mediation. By considering the entire story graph, we ensure that we never prune player action edges, i.e., we always accommodate and never intervene.

III. Story Domain

Before describing the story graph pruning process that we performed, we will introduce the story domain from our evaluation, which is also used in examples throughout this article. The domain is inspired by a subset of characters from Ware and Young's *The Best Laid Plans* [8] and realized in the *Camelot* interactive narrative sandbox tool [23].

The player begins at home, where they are given the task of acquiring a potion for their grandmother. She gives them a coin that can be used to buy an item. The game features three NPCs. A merchant is in the market selling a potion and a sword. The town guard is in the market watching for criminals. A bandit waits in her camp. The bandit has a coin that she keeps in a chest at the camp, but wants to acquire other items of value—money and potions. There are the following four locations:

1) the player's house;
2) the market;
3) the camp;
4) a crossroad that connects them all (see Fig. 2).

The game ends when the player returns home carrying the potion, or dies.

Seven actions are available. Characters can walk from one place to another. Characters can take items out of the chest in the bandit's camp. Characters can buy items from the merchant in exchange for a coin. If a character is armed, they can steal an item from an unarmed character. One character can attack and kill another, unless they are unarmed and their target is armed. Characters can loot items from slain characters. Finally, a character who knows the bandit's location can report him

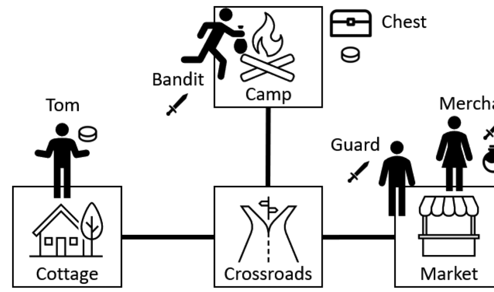


Fig. 2. A visualization of the story's initial state, showing locations and how they are connected, as well as the starting positions of each character and item

to the town guard. Despite its simplicity, this domain yields a surprising number of interesting ways the player can accomplish their goal, or die trying. There are 69 596 151 unique terminal states in the full graph (48 275 367 win states for the player, and 20 200 031 failure states).

IV. Intelligent Story Graph Pruning

In this section, we define terms relevant to story graphs and the methods we propose for pruning them. Our representation is based on Shirvani *et al.*, [24] formulation of narrative planning with intentionality and belief.

A. Story Graphs

A story domain defines objects and actions. An object is a logical constant representing a person, place, thing, or concept. Some objects are characters, intelligent agents with beliefs and goals that they plan to accomplish. An action is defined by the following four things:

1) a precondition, a conjunction of logical propositions which must be true immediately before it can occur;
2) an effect, a conjunction of logical propositions that become true immediately after the action happens;
3) one or more consenting characters who control if an action occurs, and will refuse to participate if it is not beneficial to their goals;
4) a function, $o(c, s) \in \text{true}, \text{false}$ which determines, for every character c and state s, whether character c observes the action in state s.

In short, an action defines what must be true for it to happen, what changes when it happens, who performs the action, and who observes the changes that occur. Characters have beliefs about the conditions of other objects in the domain. These beliefs may control what characters believe possible to do, or beneficial to consent to, and these are updated if they observe changes to the conditions of those objects.

Consider an action where the player gives a coin to the merchant, in the market, in exchange for the potion: $buy(Player, Potion, Merchant, Coin, Market)$. The player, coin, potion, merchant, and market are objects. The action's preconditions are that the player and merchant are alive and in the market, the player has the coin, and the merchant has the potion. The action's effects are that the player has the potion and the merchant has the coin. The player and merchant are consenting agents—both must have a reason to take that action, such as the merchant's desire for money. The observing characters for the action are any characters in the same location, so if the guard is also in the market he observes the transaction, and if the bandit is in the camp she does not, and may continue to believe that the player has the coin.

A story graph is composed of nodes representing states and directed edges representing actions. We require no particular commitment to how a state is represented, so long as a state completely specifies the configuration of the virtual world, including every character's beliefs about that configuration. The directed edges connecting states come in three types as follows.

1) A player edge is an edge for which the player is a consenting character.
2) An NPC edge is an edge for which at least one NPC is a consenting character.
3) An edge can be both a player and an NPC edge (e.g., player buys from merchant). In that case, we also call them mixed edges.

B. Experience Management

We define a full story graph as a graph containing the initial state and any state that is reachable from it by action edges.

Our goal is to begin with a full story graph, and then, prune NPC edges until the experience manager has unambiguous directions for what each NPC should do in every state. By never pruning edges that require only the consent of the player, the policy developed by the experience manager will always be able to accommodate any player action. We do allow mixed edges to be pruned, because the NPCs involved may refuse to consent to them.

It may not always be possible to avoid intervening and preventing a player action, if some sequence of player actions would lead to a dead end state. The method in this article cannot prevent this, but it can detect when it is the case. In the domain presented, no such sequences exist, and pruning is limited to NPC edges.

In a domain where actions occur instantaneously, the nodes in an unambiguous story graph would have exactly one outgoing NPC edge, or any number of outgoing player edges. In other words, in every state, the experience manager would know whether to wait for the player to act or to instruct a specific NPC to act in a specific way. However, our domain is realized in the real-time Camelot 3-D virtual environment where actions have an unknown duration.

To accommodate durations, we define an unambiguous story graph to be one where all nodes may have any number of outgoing player actions and at most one outgoing NPC action per NPC. The primary ramifications for experience management are that an action may fail or need to be canceled due to asynchronous changes to the story state. When the world transitions to a new state, our experience manager checks if there are any outgoing NPC edges for that node. If so, those NPCs are instructed to begin those actions. When an action (player or NPC) finishes, the experience manager transitions to the appropriate state in the graph, and ongoing actions are interrupted, unless that same action is also allowed in the new state, in which case the action continues.

Consider, e.g., a state where the player (who has a coin) and the bandit (who has a sword) are both at the crossroads. The experience manager must be prepared for the player to take any action, but the bandit should have clear directions to either do nothing or take a single specific action. If the bandit's directions are to rob the player, the bandit must first walk up to the player, but the player may also be moving around and performing other actions while the bandit approaches. If the player successfully executes an action during that time which makes the bandit's action impossible or undesirable (e.g., the player walks to the market), the bandit's action is interrupted, and the bandit is given new instructions based on the new state (e.g., follow the player to the market).

C. Practical Consideration: Belief

Even our small story domain can have an infinite or intractably large full story graph, depending on how one models character beliefs. Many researchers have offered models, with tradeoffs in realism and efficiency [13], [24]–[28].

We use an extremely simple model to keep the size of our domain tractable. In addition to propositions describing the physical state of the world, we track only ten specific belief propositions: the player's belief about the location of the bandit, the merchant's beliefs about the locations of the two coins, the guard's belief about the location of the bandit, the guard's beliefs about whether the player and merchant are criminals, and the bandit's beliefs about the locations of the player, the coins, and the potion. These belief propositions allow the respective characters to have wrong beliefs about certain facts. All other facts are treated as fully observable—characters cannot be mistaken about them.

D. Pruning

In this section, we explain how we prune the story graph in service of the design goals of our game, which are given as follows.

1) The NPCs should act believably.
2) The player should be active in the story.
3) The player should feel that they have agency, and they (not the experience manager) are responsible for whatever ending is achieved.
4) The game must always be finishable.

The first design goal references character believability. Determining a definition or a specific set of requirements which captures believable NPC behavior is a difficult task, as it is a perception that the audience experiences with a complex relationship to the expectations a system establishes [29]. We attempt to satisfy expectations of believability through the use of intentional agents, and the pruning steps that prioritize shorter plans and more desirable goals.

The remaining design goals seek to promote the perception of agency in players. The sense of agency arises from the ability to shape the story and the world, from opportunities to plan and act with intention, and from having expectations about this influence confirmed [6]. The design goals reflect this: We require and prioritize the player's participation in solving the problem set out by the story, we aim to clearly connect the results of the story to player action, and we avoid putting the player in a state where they cannot meaningfully effect the world and bring the story to a conclusion. We also never prune player edges, ensuring that they can always act freely within the physics of the story world.

We found the pruning criteria described in full below to work well in this domain, and we attempt to justify them by explaining our motivations and presenting illustrative anecdotes, but we do not claim they are best for all domains.

The pruning algorithm is simple: For each of the criteria described below (in the order presented), for each state node in the graph (in any order), consider the edges leading out of that state, and prune any edges that meet the criteria. Most criteria are based on the existence of paths in the graph, and since a path is a sequence of action edges taken by intentioned agents (and the player), we can think of paths as plans.

For each criteria, we include a short pseudocode snippet that describes the process. In these snippets, the following definitions are used.

1) Let $\alpha(a, v)$ denote the state after taking action a in state v. In other words, if the graph contains an action edge $v \overset{a}{\to} u$, then $\alpha(a, v) = u$, otherwise $\alpha(a, v)$ is undefined.
2) Let $\alpha(\pi, v)$ denote the state after executing a sequence of actions $\pi = \{a_1, a_2, \ldots\}$, assuming all such action edges exist in the graph.
3) Let $\beta(c, v)$ denote the state character c believes the world to be in when the world is actually in state v. For our system, we used epistemic edges to denote what state a character believes they are currently in. For a character c, if there exists an epistemic edge $v \overset{c}{\to} u$, then $\beta(c, v) = u$.
4) Let $\pi(v, a, c, g)$ denote a plan that starts with action a being taken in state v that character c believes will achieve their goal g. Formally, $\alpha(\pi(v, a, c, g), \beta(c, v)) \models g$. Note that the plan does not need to contain only actions that c consents to, because one character may expect another character to act in a helpful way. For example, the merchant can expect the player to walk to the market, where she can then sell them the medicine.

1) Intentionality Pruning: Several studies have established that intentionality, the tendency of agents to adopt and work toward goals, is an important property of believable character behavior [30], [31]. The first pruning step we apply to the story graph is to remove any NPC edges which are not intentional.[1] An action is intentional if, for every consenting character, given that character's current beliefs, there exists a sequence of causally-linked actions starting with this action that achieves a goal for that agent and such that every other action in the sequence is also explained. Due to space limitations, we must refer readers to Shirvani *et al.* [28] for full details, but here is a brief description: We say an action edge is intentional when each of its consenting characters believes that edge is a first step on a path that leads to a state where one of that character's goals becomes achieved. After this initial pruning, the full story graph contains 388 318 086 nodes connected by 1 028 110 791 edges.

To reason about intentionality and plans characters believe to be possible, the graph also must include states that characters incorrectly believe to be reachable. We model these beliefs as edges that connect states in the story to states that represent the world according to a character's beliefs. These imagined—*epistemic*—states and the actions they enable represent an additional 163 million states and 714 million edges, not included in the initial figures or future counts, which only consider story states that are reachable from the initial state by action edges. The unreachable states of the story graph are critical for NPC plans (and pruning steps that depend on them) because they model how wrong beliefs motivate actions.

Algorithm 1: Intentionality Pruning.

for all vertex v in graph **do**
  for all non-player character c **do**
    for all goal g of character c **do**
      if $\exists$ edge $v \overset{a}{\to} u$ and $\neg\exists\pi(v, a, c, g)$ **then**
        Prune $v \overset{a}{\to} u$.

2) Shorter Plan Pruning: In a state, if we can find two plans for the same agent to achieve the same goal, we prefer the shorter plan, and prune the action that begins the longer plan. These do not need to be two paths to the same state, only two paths where the same goal is achieved at the end. For example, say the guard observes the player attacking the merchant. Now the guard wants to kill the player. He could first loot the merchant's sword and then attack the player (a 2-action plan) or he could simply attack with the sword he carries (a 1-action plan). The edge where the guard picks up the merchant's sword is pruned, being the start of a longer plan for the same goal. We suggest that favoring shorter plans reduces the difficulty of discerning their goals, and may improve believably. After this pruning, the graph has 93 608 267 nodes (down 76%) and 248 440 557 edges (down 76%).

Algorithm 2: Shorter Plan Pruning.

for all vertex v in graph **do**
  for all non-player character c **do**
    for all goal g of character c **do**
      Let $v \xrightarrow{a_1} u$ be an edge s.t. $\exists\pi(v, a_1, c, g)$.
        Let $v \xrightarrow{a_2} w$ be an edge s.t. $\exists\pi(v, a_2, c, g)$.
        if $|\pi(v, a_1, c, g)| < |\pi(v, a_2, c, g)|$ **then**
          Prune $v \xrightarrow{a_2} w$.

3) Lazy NPC Pruning: In order to prioritize the player's ability to be active, we make NPC's plans more reliant on the player. Given two plans to achieve a goal, we prefer the one with more player actions. Consider the player's goal to buy the potion. The player could travel to the market, buy the potion from

[1] In practice, without limiting character actions by intentionality, the story graph is too large to generate completely. The first pruning step happens as part of the generation process. In each state, NPCs consider every reasonable plan with three or fewer actions. For each plan which achieves one of their goals, the first action in the plan is added to the graph, leading from the state being considered to the new states that follow each action. Every possible player edge is always included, and not limited by this step.

the merchant, and then return home. Alternatively, the merchant could travel to the player's home and sell them the potion without requiring the player to leave the house. Though both plans are intentional and equally short, we prefer the former, because it gives the player more opportunity to explore and find their own way to achieve their goals. It also avoids stories in which all NPCs converge on the player at the beginning, and then, constantly follow the player around, hoping for some specific interaction such as selling the potion.

We call this the *Lazy NPC principle*. Given an NPC action explained by some goal (e.g., the merchant traveling to the player's home to sell the potion), if that NPC expects the player to take an action that contributes to the same goal (e.g., the player traveling to the market to buy the potion), we prune the NPC action. After this pruning, the graph has 58 191 971 nodes (down 38%) and 148 928 950 edges (down 40%).

Algorithm 3: Lazy NPC Pruning.

for all vertex v in graph **do**
  for all non-player character c **do**
    for all goal g of character c **do**
      Let $v \xrightarrow{a_1} u$ be an edge s.t. $\exists\pi(v, a_1, c, g)$.
      Let $v \xrightarrow{a_2} w$ be an edge s.t. $\exists\pi(v, a_2, c, g)$.
      if a_1 is a player action and a_2 an NPC-only action **then**
        Prune $v \xrightarrow{a_2} w$.

4) Unique Ending Pruning: Many interactive narratives have several possible endings. In fitting with the design goal of giving the player a feeling of agency over the ending, our experience manager does not prefer any particular ending—that is to say, it is neither working with the player to achieve their goals nor working against the player to thwart them, but rather is trying to provide the ending which is a reasonable result of the player's choices.

Given two edges for the same NPC, we prune the one which most decreases the number of available types of endings. In other words, we reduce the number of situations in which NPC actions decide the ending of the story. The type of ending is either a success for the player (they are at the cottage and possess the potion), or failure (the player is not alive). Keeping open the possibility of success or failure suggests there remains an element of tension in the game, but in this scenario it also means that an NPC will not end the game by attacking the player unless there are no other viable ways to pursue their goals. This pruning step is only used to break ties, meaning it will never prune the last edge for an NPC.

Consider that the bandit wants the player's coin, and in general she can get it two ways: 1) by robbing the player; or 2) by killing the player and looting the coin. Killing the player limits the number of possible endings to 1 (the player dies), but robbing the player leaves other endings available.

However, say the player buys a sword from the merchant. Now it is impossible to rob the player, so the bandit's only way to get the coin is to kill the player. This pruning is a tie-breaker, so it will not remove the last edge for the bandit, even if it decreases the number of unique endings. We prioritize acting on one's goals over keeping endings available. Otherwise, the bandit would follow the player everywhere, always one step away from killing the player but never following through with her plan, which may harm the perception of intentionality.

It is important to prune longer plans before unique ending pruning. Consider the longer plan example above, where the guard can attack the player with his own sword (1-action) or pick up the merchant's sword and attack with that (2-action). Both plans eventually limit the story to one ending, but the first action of the 2-action plan can be taken without limiting endings. It is possible the guard will pick up the merchant's sword, leaving him with two ways to complete his goal: 1) attack with his sword; or 2) attack with the merchant's. One will be removed by unique ending pruning, but not both, since it is a tie-breaking criteria. If the attack with the merchant sword is removed, the guard will have picked up the merchant sword for no reason. Situations like this are a symptom of assuming the story graph is Markovian. Ideally, once the guard starts one plan he will continue it, but when we do not track the history of actions that brought us to the current state, the only way to know a character's "current plan" is to encode it as part of the state, which would dramatically enlarge this already intractable graph. Eventually, we intend to address this with non-Markovian experience management techniques.

Since the number of endings rendered unreachable is determined by checking the full story graph, it considers the long-term consequences of actions even if they are not clear. This is an advantage of this approach, able to detect if earlier intervention can avoid an undesirable long-term result.

Unique ending pruning targets NPC actions, not just because the experience manager tries to avoid limiting the player, but because it is possible and perhaps even desirable for the player to limit endings with their actions. When a player limits what endings are available, it can be a notable moment of agency. We plan to investigate these principles in future work.

After unique ending pruning, the graph has 52 262 059 nodes (down 10%) and 138 072 434 edges (down 7%).

Algorithm 4: Unique Ending Pruning.

for all vertex v in graph **do**
  for all non-player character c **do**
    Let $v \xrightarrow{a_1} u$ be an edge s.t. $\exists g_1$ s.t. $\pi(v, a_1, c, g_1)$.
    Let $v \xrightarrow{a_2} w$ be an edge s.t. $\exists g_2$ s.t. $\pi(v, a_2, c, g_2)$.
    Let $e_1 = 0$.
    Let $e_2 = 0$.
    for all end condition e **do**
      if $\exists$ vertex t s.t. there is a path from u to t and $t \models e$ **then**
        $e_1 = e_1 + 1$
      if $\exists$ vertex t s.t. there is a path from w to t and $t \models e$ **then**
        $e_2 = e_2 + 1$
    if $e_1 < e_2$ **then**
      Prune $v \xrightarrow{a_1} w$.

5) Player Model Pruning: In another effort to protect player agency in the story, this pruning step considers whether some

NPC choice would force the player into a certain mode of behavior. In this simple domain, the player model used is whether a player chooses to accomplish their objective as a criminal, or whether they choose to accomplish their objective without taking one of the actions that sets this flag (robbing or attacking a character who is not a criminal).

This pruning step is very similar to the unique ending step before, but focused on a different circumstance. Given two edges for the same NPC, we prune the one which most reduces the number of possible models that can still apply to the player. If an NPC action would force the player to become a criminal, and there is another action available from the same state which does not have that impact, the associated edge will be pruned. After player model pruning, the graph has 52 262 059 nodes (down 0%) and 138 072 434 edges (down 0%)—no edges were pruned as a result of this condition.

6) Goal Priority Pruning: Each agent has multiple goals to pursue, but these goals may interfere with each other and harm the perception of believability. The guard wants to kill the bandit, but he also wants to be at his post in the market. If the player reports the bandit at the crossroads, the guard will go there, and then, he has two options: 1) attack the bandit to fulfill his first goal; or 2) return to the market to fulfill his second goal. If he returns to the market, the story graph will have a cycle where the guard constantly walks back and forth between the market and crossroads, obfuscating character goals. Cycles like this are a symptom of the Markov assumption: We cannot know what plan is being acted on. This pruning provides a work-around: Agent goals are ranked by importance (in our system this is author-defined), and agents always try to complete their highest priority goal first. Killing the bandit is higher priority, so we prune the action where the guard returns to the market. In the next state, where the bandit is dead, the guard can then act on his lower priority goal of standing in the market. After goal priority pruning, the graph has 30 149 245 nodes (down 42%) and 76 006 520 edges (down 45%).

Algorithm 5: Goal Priority Pruning.

for all vertex v in graph **do**
  for all non-player character c **do**
    for all goal g_1 of character c **do**
      Let $v \xrightarrow{a_1} u$ be an edge s.t. $\exists\pi(v, a_1, c, g_1)$.
      Let g_2 be a goal for c with lower priority than g_1.
      if $\exists v \xrightarrow{a_2} w$ s.t. $\exists\pi(v, a_2, c, g_2)$ **then**
        Prune $v \xrightarrow{a_2} w$.

7) Cycle Pruning: The above prune does not prevent all cycles, so we detect cycles of three or fewer NPC action edges and break them. When an NPC has multiple actions they can take in a state, we prune those which are part of a cycle. If every edge in a cycle is that NPC's only action for that state, we prune the one which is part of the longest plan (i.e., we prefer to remove a step that requires two more steps after it to achieve the agent's goal over one that only requires one more step after it). After cycle pruning, the graph has 23 159 543 nodes (down 23%) and 56 783 502 edges (down 25%).

8) Arbitrary Pruning: If, after all of the previous steps, an NPC still has more than one action they could take in a state, we consider them equally reasonable and choose one arbitrarily. Also, we remove all outgoing edges from terminal nodes, since the game will have ended and no more actions are needed. After this final pruning, the graph has 20 365 197 nodes (down 12%) and 49 669 363 edges (down 13%).

Algorithm 6: Arbitrary Pruning.

for all vertex v in graph **do**
  if v is terminal **then**
    for all vertex v in graph **do**
      Remove all edges $v \xrightarrow{a} u$.
  else
    for all non-player character c **do**
      while $|\{v \xrightarrow{a} u : c \text{ consents to } a\}| > 1$ **do**
        Let e be an edge in $\{v \xrightarrow{a} u : c \text{ consents to } a\}$.
        Prune e.

9) Dead End Pruning: The story ends when one of the author's goals is achieved, and it must always be possible for the story to end. We define a dead end to be a node from which it is impossible to reach a terminal node—the player cannot be killed and cannot get the potion to the cottage. In this story domain, this primarily occurs when the bandit collects all the available items in the story, but does not attack the player in the process. Agents in this system act only in an intentional way, and once the bandit has collected every item she has no reason to attack the player—her goal of having all the valuable items is already fulfilled. In the unpruned full graph, these dead ends represent 56 125 nodes (0.0001% of the graph), and sampling of 1000 random stories from this full story graph indicated that 0.7% of stories encountered a dead end and could not be finished. In the final round of pruning, we remove NPC edges to ensure that no remaining dead ends are reachable. Note that we only ever remove NPC edges, never player edges; in other words, we avoid the need to ever intervene by ensuring the narrative never reaches a state where intervention might be necessary. After dead end pruning, the graph has 20 365 187 nodes (down 5%) and 49 669 351 edges (down 2%). The existence of reachable dead ends (which were present in the original graph and still present after all other pruning steps) demonstrates the need for proactive mediation which considers long-term consequences. These dead ends were not anticipated by the domain designers, and were only detected through analysis of the graph or encountering them in live play. They are the result of an interaction between the preconditions of the rob and attack actions, as well as the bandit's specific goals. In general, a connected section of dead end states in the graph may be arbitrarily large, so if an experience manager wants to guarantee that it will never intervene it may not be enough to look, e.g., only one state or only two states ahead.

V. Evaluation

We claim these pruning techniques achieve our design goals. That the story is always finishable is proved by the absence of dead ends in the final graph (i.e., from every nonterminal state these exists a path to a terminal state). We also claim these pruning techniques result in a high agency experience with believable NPC behavior, and we present the results from a play test of our game in support.

A. Experimental Design

In our survey, we compare the experience defined by our story graph to a control. The main phenomenon we want to control for is the human tendency to make narrative sense out of any sequence of events [32]. This, combined with genre expectations about adventure games, causes people to attribute intelligence to characters even when they are acting randomly. We want to demonstrate that our techniques produce believable behavior above what people would naturally perceive in this domain no matter what policy the experience manager uses. Therefore, we compare our story graph to one generated randomly. At first glance, this may seem like a weak baseline, but as we will discuss later, most people found even random NPC actions believable; they simply found ours more believable.

Like our intelligently pruned story graph, the random story graph allows every possible player action in every state. Additionally, in 75% of states, one NPC action is chosen randomly from all NPC actions possible in that state. This story graph was also pregenerated before play, so all participants experienced the same random story graph.

When we initially tested this story graph, we discovered that NPCs killed the player so frequently that it was almost impossible to achieve the ending where the player returns home with the potion. We felt this control would be too easily outperformed, so we imposed one further constraint: The simplest plan to achieve that ending (player walks to market; buys the potion; walks home) is guaranteed to be possible. Finally, to ensure it was always possible to finish the game, we perform the same dead end pruning done to the intelligent story graph. The result is a mostly random story graph in which there is at least one way to achieve both endings. It has 21 115 022 nodes and 60 492 852 edges, roughly comparable in size to our pruned graph.

We conducted a study with 20 participants, consisting mostly of undergraduate computer science students at the University of New Orleans with no prior knowledge of the game or study, who were only informed they were participating in narrative intelligence research. Participants first watched a video explaining the controls of the game, and then, completed an ingame tutorial in which they could become directly familiar with the controls, locations, and characters. In the tutorial characters take no actions, but introduce themselves and their goals through dialog when the player interacts with them. Initially, we ran the study and collected data, but results offered no insight into the questions asked: We observed that players significantly preferred whichever version they played first, regardless of treatment. We attributed this to the novelty of exploring the virtual world and created the tutorial in response to this, then ran the study again. The tutorial ensures that participants have explored the world before playing the game, allowing them to focus on the narrative.

After the tutorial, each participant played two versions of the game: One using the random story graph and the other using the intelligent story graph produced by our pruning. Participants were randomly divided into two groups, with one playing the random version first and the other playing the intelligent version first. Participants were required to complete each version twice (to ensure they had a chance to try different strategies), but were invited to play up to ten times. We did not require them to win or to experience different endings.

B. Results

After playing the two versions, participants were shown four statements about character believability and agency, and were asked to choose whether they felt the statement was more agreeable for the first version, or the second. Table I presents the breakdown of results by statement, showing the numbers of participants who preferred the intelligently pruned version or the random version.

We hypothesize that players will significantly prefer the intelligent story graph, i.e., they will say these statements were more true of the intelligent story graph. A binomial exact test confirmed this hypothesis for three of the four questions at the $p < 0.05$ level. The p-values in Table I are given after applying Benjamini and Hochberg's [33] correction for multiple hypothesis testing. Effect size is given as relative risk that participants preferred the random version.

We did not detect a significant effect at the $p < 0.05$ level for the statement, "The characters reacted to things they saw and ignored things they did not see." We believe the results, a roughly 2:1 ratio in favor of the pruned version, are suggestive of a trend toward that preference and must be investigated with a greater number of participants.

VI. Practical Consideration: Scalability

The system presented represents a method of proactive mediation where all possible stories are considered, and a curated subset of stories is retained. In practice, the method as described in this article is limited by computational cost, especially the cost with respect to the space needed for the full story graph. However, this cost can be mitigated. One method for mitigation was used for the system described above, as generation of the full story graph without intentionality proved to be impossible at this level of story complexity. To overcome this limit, intentionality was treated as a necessary precondition to generation, rather than a true pruning step. All other steps were implemented after this initial generation so that multiple orderings of the given strategies could be explored and developed upon. In practice, the pruning steps could become conditions for generation, or could be implemented as part of an iterative prune-then-deepen process, alleviating some limits on the underlying story complexity.

The pruned graph developed by the system occupies 3 GB of memory in a plaintext format, for which responsive graph navigation requires several MBs of overhead. These demands may be unsuitable for some game environments or embedded systems. Here, we indicate how the graph may be compressed while maintaining usability. Translating the plaintext format into a straightforward packed binary format reduces the space

TABLE I
SURVEY RESPONSES FROM 20 PARTICIPANTS

Statement	Prefer pruned	Prefer random	p-value (corrected)	Relative risk
The characters felt realistic.	16	4	**0.0079**	0.4
The characters reacted to things they saw and ignored things they did not see.	13	7	0.1316	0.7
The characters tried to accomplish their goals.	18	2	**0.0008**	0.2
My actions had a significant effect on the story.	16	4	**0.0079**	0.4

requirement to 1.6 GB of space, and allows fast graph navigation with only KBs of overhead.

For more extensive compression of this representation, we adapt the experience management results to a policy function. We make the assumption that the game environment itself implicitly includes the possible states and available player actions by encoding action preconditions and effects, and only consider the NPC actions from each state. In each state of the pruned graph, for each NPC, there is either one action or no action—which we represent as the no-op action *wait(c)*. Each state is unique, and associated with a single result, representing a policy function $q(s, c) \to a$ that determines, for some state s and some NPC c, the action a that c will try to take. This function can be modeled using machine learning methods like decision trees or neural networks, yielding a compressed version of the pruned graph.

A decision tree uses 40 MB of memory and takes only a few minutes to train, resulting in a model with 100% accuracy for replicating the action an agent would take in the pruned graph. Neural networks can provide further compression. Using a model with five linear hidden layers of 64 dimensions each, 360 KB between all the networks, it took our models a few minutes to reach 95% accuracy, and after an hour each model exceeded 99% accuracy. Reaching 100% accuracy is a more time consuming prospect—training for the bandit's actions took 2 days, and for the guard it took 5 days and 17 hours. At time of writing, the merchant's training did not reach 100%.

These compressed models can embed the pruned graph in another system, looking up actions from the experience manager policy in real-time with minimal overhead. They also offer the potential to study the policy using the architecture of these models—in the case of the decision tree—and machine learning techniques for explaining model behavior.

VII. CONCLUSION

In this article, we frame experience management as a story graph pruning problem. By starting with a full story graph and pruning only NPC actions, we precompute the experience manager's policy, accounting for the long-term effects of those decisions on the entire space of possible stories. We ensure NPCs act believably and that the story can always reach an ending while also ensuring the experience manager never needs to prevent the player from taking an action.

We learned several important lessons from this work. First, story graphs, even for small domains, can get very big very fast. Even in our simple domain, when limiting NPC plans to three steps and accounting for only nine beliefs, the graph contains over 300 million state nodes and 1 billion edges, and that number does not count the states which characters believe to be possible but are actually impossible. Pruning a complete story graph will be intractable for most domains, but this work was instructive because it allowed us to consider the long-term consequences of every experience manager decision. We believe these insights can be applied, probably as heuristics, to larger graphs which must be generated on demand.

The second lesson is: In storytelling domains like this one random actions are a surprisingly strong baseline. After playing the first version of the game (but before playing the second), participants responded to the four statements in Table I on a five point Likert scale. Two groups of ten participants were not a large enough sample for a between subjects analysis, but both groups tended to agree with all four statements, even those who played the random story graph. Anecdotally, several participants invented elaborate explanations to make sense of the random actions they saw and said they enjoyed these "plot twists." The human tendency to narrativize events [32] may be so strong that most people cannot see actions as random, only as easier or harder to explain, and thus, a randomly generated story graph is a stronger baseline than it might seem.

We feel that the most limiting assumption of this initial work is that the story graph is Markovian. Stories are non-Markovian; different action sequences leading to the same state often require different conclusions. In future work, we intend to explore how tracking the history of events can improve experience management and NPC believability.

APPENDIX
STORY GRAPH ARTIFACT

To make this work reproducible and to encourage others to experiment with story graph pruning criteria, we have released the story graphs described in this article.[2]

ACKNOWLEDGMENT

Supercomputing was provided by the Louisiana Optical Networking Initiative. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] M. O. Riedl and V. Bulitko, "Interactive narrative: An intelligent systems approach," *AI Mag.*, vol. 34, no. 1, pp. 67–77, 2013.
[2] B. Games, "Dwarf fortress," 2002. Accessed: Jul. 2022. [Online]. Available: http://www.bay12games.com/dwarves/
[3] R. H. Stottler and M. Vinkavich, "Tactical action officer intelligent tutoring system (TAO ITS)," Defense Technical Information Center, Virginia, USA, Tech. Rep., 2006.

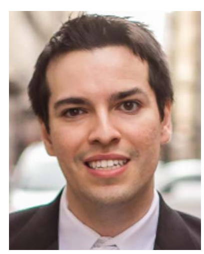
[2] [Online]. Available: http://cs.uky.edu/%7sgware/projects/storygraphs

[4] T. Games, "The walking dead," 2012. Accessed: Jul. 2022. [Online]. Available: https://store.steampowered.com/app/207610/The_Walking_Dead/
[5] S. Chen, M. J. Nelson, A. Sullivan, and M. Mateas, "Evaluating the authorial leverage of drama management," in *Proc. AAAI Spring Symp. Intell. Narrative Technol.*, 2009, pp. 20–23.
[6] N. Wardrip-Fruin, M. Mateas, S. Dow, and S. Sali, "Agency reconsidered," in *Proc. Digit. Games Res. Assoc.*, 2009. [Online]. Available: http://www.digra.org/digital-library/forums/5-breaking-new-ground/
[7] M. Mateas and A. Stern, "Structuring content in the Façade interactive drama architecture," in *Proc. 1st AAAI Int. Conf. Artif. Intell. Interactive Digit. Entertainment*, 2005, pp. 93–98.
[8] S. G. Ware and R. M. Young, "Intentionality and conflict in the best laid plans interactive narrative virtual environment," *IEEE Trans. Comput. Intell. Artif. Intell. Games*, vol. 8, no. 4, pp. 402–411, Dec. 2016.
[9] M. O. Riedl and R. M. Young, "From linear story generation to branching story graphs," *IEEE Comput. Graph. Appl.*, vol. 26, no. 3, pp. 23–31, May/Jun. 2006.
[10] S. G. Ware, E. T. Garcia, A. Shirvani, and R. Farrell, "Multi-agent narrative experience management as story graph pruning," in *Proc. 15th AAAI Int. Conf. Artif. Intell. Interactive Digit. Entertainment*, 2019, pp. 87–93.
[11] D. Thue and E. Carstensdottir, "Getting to the point: Resolving ambiguity in intelligent narrative technologies," in *Proc. Joint Workshop Intell. Narrative Technol. Workshop Intell. Cinematogr. Editing 14th AAAI Int. Conf. Artif. Intell. Interactive Digit. Entertainment*, 2018. [Online]. Available: http://ceur-ws.org/Vol-2321/
[12] C. Swinehart, "One book, many readings," 2009. Accessed: Jul. 2022. [Online]. Available: http://samizdat.cc/cyoa/
[13] J. Bates, "Virtual reality, art, and entertainment," *Presence Teleoperators Virtual Environ.*, vol. 1, no. 1, pp. 133–138, 1992.
[14] P. W. Weyhrauch, "Guiding interactive drama," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, USA, 1997.
[15] M. J. Nelson, M. Mateas, D. L. Roberts, and C. L. Isbell, "Declarative optimization-based drama management in interactive fiction," *IEEE Comput. Graph. Appl.*, vol. 26, no. 3, pp. 32–41, May/Jun. 2006.
[16] D. L. Roberts, S. Bhat, K. S. Clair, and C. L. Isbell Jr, "Authorial idioms for target distributions in TTD-MDPs," in *Proc. 22nd Conf. Assoc. Advance. Artif. Intell.*, 2007, pp. 852–857.
[17] D. Thue and V. Bulitko, "Procedural game adaptation: Framing experience management as changing an MDP," in *Proc. 8th AAAI Int. Conf. Artif. Intell. Interactive Digit. Entertainment*, 2012, pp. 44–50.
[18] M. Arinbjarnar, H. Barber, and D. Kudenko, "A critical review of interactive drama systems," in *Proc. AI Games Symp. Adaptive Emergent Behav. Complex Syst. Conv.*, 2009, pp. 35–46.
[19] R. Farrell, S. G. Ware, and L. J. Baker, "Manipulating narrative salience in interactive stories using indexter's pairwise event salience hypothesis," *IEEE Trans. Games*, vol. 12, no. 1, pp. 74–85, Mar. 2020.
[20] B. Kybartas and R. Bidarra, "A survey on story generation techniques for authoring computational narratives," *IEEE Trans. Comput. Intell. Artif. Intell. Games*, vol. 9, no. 3, pp. 239–253, Sep. 2016.
[21] M. Riedl, C. J. Saretto, and R. M. Young, "Managing interaction between users and agents in a multi-agent storytelling environment," in *Proc. 2nd Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2003, pp. 741–748.
[22] J. Harris and R. M. Young, "Proactive mediation in plan-based narrative environments," *IEEE Trans. Comput. Intell. Artif. Intell. Games*, vol. 1, no. 3, pp. 233–244, Sep. 2009.
[23] A. Shirvani and S. G. Ware, "Camelot: A modular customizable sandbox for visualizing interactive narratives," in *Proc. 12th Intell. Narrative Technol. Workshop 16th AAAI Int. Conf. Artif. Intell. Interactive Digit. Entertainment*, 2020.
[24] A. Shirvani, S. G. Ware, and R. Farrell, "A possible worlds model of belief for state-space narrative planning," in *Proc. 13th AAAI Int. Conf. Artif. Intell. Interactive Digit. Entertainment*, 2017, pp. 101–107.
[25] J. Porteous, M. Cavazza, and F. Charles, "Applying planning to interactive storytelling: Narrative control using state constraints," *ACM Trans. Intell. Syst. Technol.*, vol. 1, no. 2, pp. 1–21, 2010.
[26] J. Ryan, A. Summerville, M. Mateas, and N. Wardrip-Fruin, "Toward characters who observe, tell, misremember, and lie," *Proc. AAAI Int. Conf. Artif. Intell. Interactive Digit. Entertainment*, vol. 11, no. 1, pp. 56–62, 2015.
[27] M. Eger and C. Martens, "Character beliefs in story generation," in *Proc. 13th Artif. Intell. Interactive Digit. Entertainment Conf.*, 2017, pp. 184–190.
[28] A. Shirvani, R. Farrell, and S. G. Ware, "Combining intentionality and belief: Revisiting believable character plans," in *Proc. 14th AAAI Int. Conf. Artif. Intell. Interactive Digit. Entertainment*, 2018, pp. 222–228.
[29] T. J. Tanenbaum and J. Bizzocch, "Close reading Oblivion: Character believability and intelligent personalization in games," *Loading...*, vol. 3, no. 4, 2009. [Online]. Available: https://journals.sfu.ca/loading/index.php/loading/issue/view/5
[30] M. O. Riedl and R. M. Young, "Narrative planning: Balancing plot and character," *J. Artif. Intell. Res.*, vol. 39, no. 1, pp. 217–268, 2010.
[31] S. G. Ware, "A plan-based model of conflict for narrative reasoning and generation," Ph.D. dissertation, North Carolina State Univ., Raleigh, NC, USA, 2014.
[32] J. Bruner, *The Narrative Construction of Reality*. Washington, DC, USA: American Psychological Assoc., 1991, pp. 1–21.
[33] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *J. Roy. Stat. Soc. Ser. B. Methodological*, vol. 57, no. 1, pp. 289–300, 1995.

Stephen Ware (Member, IEEE) received the B.S. degree in both computer science and philosophy from Loyola University, New Orleans, LA, USA, in 2008, and the M.S. and Ph.D. degrees in computer science from North Carolina State University, Raleigh, NC, USA, in 2011 and 2014, respectively.

He is currently an Assistant Professor of computer science and the Director of the Narrative Intelligence Lab, University of Kentucky, Lexington, KY, USA. His research interests include computer narrative, especially automated planning, knowledge representation, narratology, cognitive science, and artificial intelligence in games.

Dr. Ware is a Member of the Association for the Advancement of Artificial Intelligence, the International Game Developers Association, and the Association for Computing Machinery.



Edward T. Garcia received the B.S. and M.S. degrees in computer science from the University of New Orleans, New Orleans, LA, USA, in 2005 and 2019, respectively.

He developed methods for automatic experience management in Camelot and in a virtual reality Police Use of Force training simulation. His research interests include game development and artificial intelligence.

Mira Fisher received the B.S. degree in computer science from the University of Kentucky, Lexington, KY, USA, in 2015, where she is currently working toward the Ph.D. degree, in computer science with the University of Kentucky, Lexington, KY, USA.

She is currently a Research Assistant with the University of Kentucky. Her research interests include interactive narrative systems, automated planning, and machine learning.

Alireza Shirvani received the M.S. degree in computer engineering from the University of Isfahan, Isfahan, Iran, , in 2014; and the Ph.D. degree in computer science from the University of Kentucky, Lexington, KY, USA, in 2022.

He is currently an Assistant Professor with Saint Louis University, Saint Louis, MO, USA, and is now working on a sandbox virtual environment for intelligent drama management. His research interests include artificial intelligence, affective computing, and procedural content generation.

Rachelyn Farrell received the B.S. degree in computer science from the University of Mississippi, Oxford, MS, USA, in 2012, and the M.S. degree in computer science from the University of New Orleans, New Orleans, LA, USA, in 2017. She is currently working toward the Ph.D. degree in computer science with the University of Kentucky, Lexington, KY, USA.

She is currently a Research Assistant with the University of Kentucky. Her research interests include computational narrative, automated planning, interactive narrative systems, and virtual environments.