Intelligent De-Escalation Training via Emotion-Inspired Narrative Planning

Mira Fisher[1], Cory Siler[1] and Stephen G. Ware[1]

[1]Narrative Intelligence Lab, Department of Computer Science, University of Kentucky, Lexington, KY, USA 40506

Abstract

We present an intelligent experience management architecture for a virtual reality police de-escalation training platform we are currently developing. Our aim is to direct the cast of non-player characters toward a scenario outcome appropriate to the player's decisions, resulting in bad endings precisely when player's mistakes enable them. We use a narrative planner to generate a story graph representing every possible narrative, and then we prune the graph to eliminate less believable non-player character actions. Unlike previous approaches based on story graph pruning, we implement an emotional planning model that lets us represent characters acting out of fear of bad outcomes as well as hope for good ones. We also incorporate experience management techniques for delaying commitment to hidden settings of the scenario and for capitalizing on player mistakes to demonstrate the negative consequences of not following best practices.

Keywords

emotional planning, experience management, intelligent training systems, narrative planning

1. Introduction

Planning-based narrative technologies are promising for intelligent training systems because they can adapt to a wide variety of player behaviors while shaping the player's experience to meet pedagogical goals. We are exploring police de-escalation training as a potential application for recent advances in intelligent narrative. The purpose of police de-escalation training is to help habituate officers to defusing tense situations, avoiding violence when possible. This application is a meaningful testbed in part because of the type of character modeling needed. When real police encounters with civilians end violently, an often-cited factor is the parties' mental models of each other—e.g., one party reacting aggressively based on their own belief that the other party intends to harm them. We model lines of reasoning like this in our system's non-player characters (NPCs).

In our system, the player takes the role of a police officer who has just pulled over a car carrying the driver and a passenger, both NPCs. The player's objective is nominally to issue a traffic citation to the driver, but a variety of problems can arise depending on hidden settings that the system chooses adaptively. Looking up the driver's ID in the police database may or may not reveal that the driver has a restraining order against someone. The person on the restraining order may or may not be the passenger in the car, which the player can only find out for certain by interacting with the passenger to get their ID. If the restraining order is against the passenger and the player fails to investigate, the driver could be in danger; alternatively, the player risks a civil liberties violation or outright violence by demanding the passenger's ID. The scenario ends when the traffic stop is completed, the player arrests an NPC, or the player or an NPC is harmed; the player is presented with an ending scene based on the outcome.

Figure 1: The virtual reality environment.

The training simulation is realized in a room-scale virtual reality environment, pictured in Figure 1. The player can walk around the virtual space and interact with virtual props in the environment: their gun and handcuffs, characters' IDs and a laptop for looking up the IDs in a database, the traffic citation, etc. NPCs act only when issued commands by the *experience manager*, a disembodied intelligent agent that directs the story based on its pedagogic goals.

This paper focuses on the experience manager, which makes these contributions:

13th Intelligent Narrative Technologies Workshop, October 25, 2022, Pomona, CA

mira.fisher@uky.edu (M. Fisher); cory.siler@uky.edu (C. Siler); sgware@cs.uky.edu (S. G. Ware)

- We extend previous work on narrative planning to reason about joy, fear, and relief, similar to Shirvani and Ware [1], to leverage each character's theory of mind and model more realistic behavior.
- Some features of the world state are initially unobserved by the player, and the experience manager can determine them during play to bring about its desired ending, so long as these decisons do not violate the player's observations, similar to work by Robertson and Young [2].
- The experience manager's goal is to bring about a bad ending only when a player's actions enable it, demonstrating the possible negative consequences of failing to follow best practices.

For example, consider an instance of the scenario where the player gets the driver's ID and looks it up in the database while printing the traffic citation. They realize the driver has a restraining order against someone, possibly the passenger. The player could preemptively arrest the passenger out of fear, but the experience manager can then decide that the passenger was just an innocent civilian, demonstrating the danger of making an arrest before gathering enough information.

Our evaluation highlights the difficulty of performing quality assurance on a large story space. Though our scenario is relatively simple, we hope to one day enable story spaces too large for a human author to anticipate in advance. How, then, can we ensure all narratives meet the author's pedagogic goals? We make a first attempt at this by sampling possible narratives from our experience manager in hopes of identifying a positive correlation between both good endings and actions we wish to encourage, and between bad endings and actions we wish to discourage.

Before we proceed, in the spirit of the call by Martens and Smith [3] "to pair the creation of [narrative artificial intelligence] with critical reflection on the underlying (often implicit) metaphors and values used by its creators", and particularly their discussion about the reductive nature of systems built around the creator's definition of social believability, we acknowledge the inherent limitations of our system (or any system) as a model of real police encounters. No mental model we build our NPCs around will ever fully capture the psychological nuances of a real situation, especially around factors like racism that play an important role in discussions of police use of force. No utility function or ranking of outcomes will truly do justice to the ethical values at stake. We hope that our system will one day serve as a helpful tool for part of the de-escalation training process, but we make no claim that it will serve as a complete de-escalation training curriculum, just as improvements to police de-escalation training overall may help reduce violence but are just one part of the conversation around the role of policing in our society.

2. Related Work

Interactive training and tutoring systems guided by artificial intelligence can yield significant gains in learner performance over traditional instruction methods [4]. De-escalation training as an application has been explored before. Bosse et al. [5] present a virtual reality de-escalation training platform for public transport employees dealing with aggressive passengers. The player's dialogue with NPCs is governed by conversation trees; the NPCs exhibit different forms of aggression and the player is tasked with choosing a response that shows the appropriate communication style to defuse a given form of aggression. Bosse and Gerritsen [6] adapt this system for police academy students with a scenario about responding to a call about domestic violence.

For intelligent interactive narratives, including but not limited to many training applications, planning is an effective tool because it offers a formal, generative model of a sequence of actions. We draw on intentional planning [7, 8, 9], which extends classical planning to ensure NPCs give the appearance of pursuing their own individual goals. We also draw on later extensions that enable NPCs to have theory of mind and wrong beliefs [10], and to act based on a model of emotion that explains actions in terms of joy, hope, fear, and relief [1].

Planning in general has been used in a variety of intelligent training and tutoring systems; see Cogollo et al. [11] for an extensive survey. The system by Ramírez and De Antonio [12] uses an experience manager with two components: one that plans an ideal solution to a problem, and one that monitors a trainee's adherence to that solution or the possibility that the trainee's deviations lead to an alternative solution. The system by Vannaprathip et al. [13] encodes procedural knowledge as a PDDL planning domain and generates questions about hypotheticals ("What if you had...") and rationales ("Why did you...") to check a trainee's understanding of their own decisions as they complete a task. The system by Thomas and Young [14]'s uses a plan-space representation to model the ways a trainee might plan for tasks within the game world and how the training agent can reveal flaws in the trainee's plan. The role of planning in all three of these systems is to encode how a task might be done, so that this knowledge can be transferred to a user; our system specifically highlights the negative consequences of failing to follow best practices.

The study of experience management has an extensive history [15], ranging from experience managers that occasionally influence mostly-autonomous NPCs to architectures like ours where NPC actions are entirely dependent

on the experience manager. By modeling a playthrough as a joint traversal of a graph by the experience manager and player, with the experience manager trying to optimize for a particular definition of story quality, our approach falls under the Search Based Drama Management [16] paradigm.

The most similar system to ours is that of Garcia et al. [17], which focused on measuring trainees' immersion in an intelligent virtual reality training simulation. We share their core approach of story-graph-based experience management [18], but we extend the model of narrative to include emotion, allow the experience manager to set unobserved features of the state, and evaluate a specific theory of which ending the experience manager should attempt to bring about.

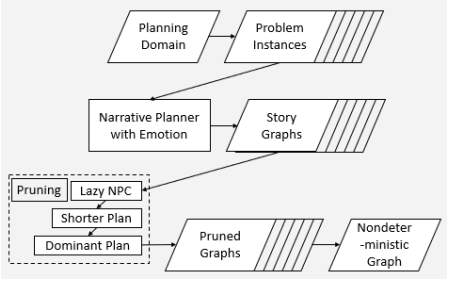

Figure 2: The offline process for generating the graph that the experience manager will use online to make decisions during a playthrough.

3. Approach

We consider experience management via the generation and pruning of a *story graph* [19], a directed graph where nodes represent states of the story world and edges show state transitions resulting from player or NPC actions. The pipeline for generating the final graph is illustrated in Figure 2. First, we represent the high-level events that could occur in our training simulation as a narrative planning domain and use a modified version of the Sabre planner [20] to enumerate the resulting story graph. We prune NPC actions from the story graph down to a believable subset of these actions using methods based on techniques by Ware et al. [18]. Then, given multiple story graphs with variations on the initial state of the scenario, we combine these story graphs into a nondeterministic story graph which tracks the different possible worlds the player could be in based on their prior observations. Due to the hidden nature of the initial state variables, the experience manager need not commit right away to which is the "real" scenario, but the construction of the nondeterministic graph ensures that observed NPC behavior prior to that commitment will be consistent with that scenario.

During play of the scenarios, the experience manager agent and the player jointly traverse the nondeterministic pruned graph until they reach a terminal state. The experience manager's goal is to bring about the worst ending that a player's choice makes available to it.

3.1. Narrative Planning Model

We give a high-level overview here of the planning model we use to define the states and the valid transitions between them that make up the story graph. It is a superset of the belief-intention planning model by Ware and Siler [20] and a subset of the emotional planning model by Shirvani and Ware [1], with minor modifications we mention in this section; full details of the formalism can be found in those papers.

A narrative planning domain defines variables that describe the story world, such as the location and status of all objects. It also defines a set C of characters, special objects which can have beliefs and intentions.

A node s in a story graph is a world state, which is any function that can determine whether a Boolean logical proposition is true or false. States track the value currently assigned to each variable as well as each characters' beliefs. A state can answer whether the Passenger is armed, whether the Officer believes that the Passenger is armed, whether the Passenger believes that the Officer believes that the Passenger is armed, etc. When the world is in state s, we use $\beta(c, s)$ to denote the state character c believes the world to be in. The details of how beliefs are handled is not directly relevant to this paper, so we refer readers to Ware and Siler [20].

A narrative planning domain defines actions that can change the world state. Actions are based on classical STRIPS-like planning [21] with some additions. Every action defines $\textsc{pre}(a)$, a logical proposition which must be true in the state before it occurs, and $\textsc{eff}(a)$, a logical proposition that must be true in the state after it occurs. For narrative planning, every action a also defines a set $\textsc{con}(a)$ of characters $\in C$ who must have a reason to take the action. Actions also define how character beliefs change as a result, and we omit those details here. In short, if a character observes an action their beliefs are updated, and otherwise their beliefs stay the same.

A story graph may have an edge $s_1 \xrightarrow{a} s_2$ from node s_1 to node s_2 via action a if $\textsc{pre}(a)$ is true in s_1 and s_2 is the state that would result from taking action a. We use $\alpha(a, s)$ to mean that state after taking action a in state s. So when a graph has an edge $s_1 \xrightarrow{a} s_2$, then $\alpha(a, s_1) = s_2$. If a's precondition is not true in s, $\alpha(a, s)$ is undefined. We also use $\alpha(\{a_1, a_2, ..., a_n\}, s)$ to denote the state after taking the sequence of actions $\{a_1, a_2, ..., a_n\}$.

Our narrative planner aims to achieve the author's goals for the story by only taking actions that are explained by the emotions of the characters who take them. We use utility functions to reason about three basic emotions: joy, fear, and relief. Agents feel joy when a plan increases their utility, they fear plans that decrease their utility, and they feel relief when a plan prevents a fear. To define these, we need to distinguish between explained actions (that characters want to happen) and expected actions (that characters think can happen, regardless of whether they want it).

Utility functions map world states to real numbers. Let $U(s)$ be the author's utility in state s, and for every character $c \in C$ let $U_c(s)$ be c's utility in state s.
An action a_1 is *expected* by character c in state s just when:

1. there exists a sequence of actions $\pi = \{a_1, a_2, ..., a_n\}$ such that
2. $\alpha(\pi, \beta(c, s))$ is defined and
3. every action a_i is explained in state $\alpha(\{a_1, a_2, ..., a_{i-1}\}, s)$

In other words, a character c can expect an action when (1) it is the first in a sequence of one or more actions (2) that the character believes can occur and (3) every action in that sequence makes sense for all characters involved.

Explained actions are expected actions that cause one of the positive emotions, joy or relief. An action a_1 is *explained by joy for* character c iff there exists an expected sequence of actions π that begins with a_1 and:

1. $\pi = \{a_1, a_2, ..., a_n\}$ is an expected sequence of actions such that
2. $U_c(\alpha(\pi, \beta(c, s))) > U_c(\beta(c, s))$, and
3. no strict subsequence of π exists that also meets these criteria.

In other words, an action is explained by joy if a character would take that action as part of a plan they believe will increase their utility, and the plan contains no unnecessary actions. In our domain, the Officer's and Driver's utilities are highest once the traffic stop is safely concluded, so the Driver will give their ID to the Officer, and the Officer will return the ID with a traffic citation in pursuit of that higher utility.

To define relief, we must first define fear. A character c in state s *fears* action a_1 will lead to utility u iff there exists an expected sequence of actions π_{fear} such that $U_c(\alpha(\pi_{fear}, \beta(c, s))) = u$ and $u < U_c(\beta(c, s))$. In other words, the character believes the action can lead to a lower utility u. The Officer's utility is low if an innocent person is dead, so when the Officer believes there is a restraining order against the Passenger and the Passenger is armed, the Officer can fear the Passenger will hurt the Driver.
An action a_1 is *explained by relief for* character c in state s iff:

1. c fears π_{fear} will lead to utility u and
2. $\pi_{relief} = \{a_1, a_2, ..., a_n\}$ is an expected sequence of actions such that
3. in state $\alpha(\pi_{relief}, \beta(c, s))$ there does not exist an expected sequence π'_{fear} that would lead to utility u, and
4. no strict subsequence of π_{relief} exists that also meets these criteria.

For example, if the Officer fears the Passenger will hurt the Driver, the Officer can arrest the Passenger to relieve that fear.

Now that we have defined when an action is explained for a character by joy or relief, we say that an action a is *explained* (in general) when it is explained for every consenting character $c \in \text{CON}(a)$. A valid story is any sequence of explained actions that increases the author's utility. That is, for every action in that story, for every character who takes that action, we can identify a source either of joy or of relief that motivated the action.

One character, the Officer in our domain, is labeled as the player character and is an exception to some of our constraints on character behavior. The planner still tracks a set of beliefs and assumes a utility function for the Officer, and they are named as a consenting character in a number of actions. Modeling the player character in this way is necessary to model NPC behavior, which accounts for plans that NPCs believe to be explained according to the Officer's utility function, and for modeling the Officer's expected beliefs about the state of the world. However, the player is free to have the Officer character take any action, regardless of whether that action is explained for the Officer according to the model.

3.2. Narrative Planning Domain

The domain we designed for this prototype was created in consultation with Jennifer Melgar, at that time a police officer who organized de-escalation training role-playing exercises to train officers at the University of Kentucky Police Department. It is based on one of her real experiences and a discussion of hypothetical alternatives that could have occurred. We describe a partial implementation here, which we plan to update in later iterations as we improve the scalability of our story graph generation process.

There are three characters: the Officer (player), the Driver of the car, and a Passenger. The scenario begins after the Officer has pulled the Driver over for erratic

driving. Objects include the Officer's handcuffs and gun, ID cards for the Driver and Passenger, a computer in the Officer's vehicle, a citation printer, a traffic citation, and possibly a gun hidden by the Passenger.

There are three features of the domain which can vary: whether the Driver has a restraining order against *someone*, whether the Driver has a restraining order against the *Passenger specifically*, and whether the Passenger has a gun.
These actions are possible:

- One character can give an item to another; both giver and receiver are consenting characters.
- The Officer can use the computer to look up the Driver's ID while in possession of it. This creates a traffic citation and reveals whether the Driver has a restraining order (without revealing whether the target is the Passenger). To determine whether the target is the Passenger, the Officer needs to see the Passenger's ID.
- If the Officer knows about a restraining order, they can explain to the Passenger why they want to see the Passenger's ID. This makes the Passenger believe the *Officer* believes two things, regardless of whether they are actually true or whether the Officer actually knows them: that the Passenger is the target of the restraining order and that the Passenger is armed. (Although this action affects only character beliefs and not the "real" values of any variables, it can motivate the Passenger's future behavior, e.g., to try to prove their innocence or avoid arrest.)
- The Officer can arrest another character.
- One character can shoot another if the shooter has a gun.

The author's utility function ranks endings as:

1. Worst: An innocent person is killed.
2. An innocent person is arrested.
3. There is a restraining order against the Passenger, and the Passenger was killed.
4. There is a restraining order against the passenger, and the Passenger was arrested.
5. Best: There is no restraining order against the Passenger, and the Driver's ID has been returned to the Driver (with or without the citation).

Character utility functions use similar reasoning, except that characters consider bad things happening to themselves worse than to others. For example, the Driver ranks an ending where they are killed as worse than an ending where some other innocent character is killed.

The Officer, Driver, and Passenger all want the traffic stop to be over (caused by returning the Driver's ID). When there is a restraining order against the Passenger, the Driver wants the Passenger to be arrested or dead, and the Passenger wants the Driver to be dead.
This scenario is designed to explore several possibilities:

- If the Driver is in danger, the Driver may have been trying to get an officer's attention on purpose.
- If the Driver may be in danger, it is the Officer's responsibility to learn more and keep the Driver safe.
- Because the passenger was not driving, there is no apparent obligation for them to give their ID to the Officer. Demanding the passenger's ID without explaining the need for it may be a violation of the Passenger's civil liberties.
- The Officer should explain their concerns and their request for the Passenger's ID to the Passenger, rather than demanding it, taking it by force, or preemptively arresting the Passenger.
- The Officer should be prepared in case the Passenger is a danger to the Driver and in case they are armed.

3.3. Story Graph Generation

We begin by generating a full story graph, which includes every possible state that could ever occur and every allowable edge. Story graphs can be infinite, but we designed our domain to yield a finite graph that is small enough to generate in approximately 12 hours on a modern desktop computer.

3.4. Story Graph Pruning

Given a full story graph of reachable states with all legal actions, we adapt methods from Ware et al. [18] to prune the graph, i.e., remove NPC actions from the graph to improve the overall quality of the set of possible player experiences. We never remove actions that require only player consent during pruning, because the experience manager needs to be prepared to respond to any action taken by the player. Some actions (like the Passenger giving their ID to the Officer) require both NPC and player consent; these may be removed if the NPCs do not have a good reason to take them.

Below we describe several pruning techniques in the order they were applied.

During *intentionality pruning*, we remove any edge that is not explained for an NPC who is a consenting character. This step is more inclusive about which edges survive than prior models, such as that of Ware et al. [18], because an NPC can consent to an action if they believe it can eventually lead to a utility increase *or* to a state that prevents something they fear.

The original Ware et al. [18] definitions for the remaining pruning methods assume that each action is paired with a single explanation for each consenting character; the pruning methods are based on comparing two actions and their associated single explanations. Because our system generates all possible explanations for each action, we redefine the pruning methods in terms of pruning *explanations*; if an action has had all of its explanations pruned for a given NPC, we prune that action edge.

With *lazy NPC pruning*, we bias the graph to favor stories where the player takes a more active role. We consider all explanations available to an NPC in a given state. We prune an explanation if there exists another explanation considered by the same NPC, from the same state, with the same expected joy and relief, but which contains a greater number of actions requiring the player's consent. In other words, an NPC will not act without the player if they expect the player to work toward the same result. For example, the Driver will not get out of the car to bring their ID to the Officer when they expect the Officer to come and ask for it.

With *shorter plan pruning*, we encourage NPCs to act efficiently in pursuit of their goals. We consider all explanations available to an NPC in a given state. We prune an explanation if there exists another explanation considered by the same NPC, from the same state, with the same expected joy and relief, but which requires fewer actions to be realized. For example, the Driver could give their ID to the Passenger and let the Passenger hand the Driver's ID to the Officer, but this is an unnecessarily long plan, and the Driver will prefer to simply hand their ID directly to the Officer instead.

With *dominant plan pruning*, the counterpart to *goal priority pruning* from Ware et al. [18], we prevent NPCs from pursuing an outcome when a strictly better outcome is possible. Borrowing from the field of multiobjectve optimization, we consider one explanation to *dominate* another for an NPC if it results in either higher joy or higher relief while being no worse in the other emotion. Among the set of all explanations available to an NPC for actions in a given state, we prune explanations that are dominated by another by this definition. For example, suppose the Passenger is innocent but they believe the Officer suspects them of restraining order violation and they expect that the Officer may shoot them. The Passenger could get relief from the fear of getting shot from either a plan to get arrested instead, or a plan to show their ID proving they are not on the restraining order. The latter plan causes greater relief, without being worse in terms of joy, so the former is marked as dominated and pruned.

3.5. Nondeterministic Story Graph

We aim to give the player the impression of a predetermined story world. For instance, from the Officer player's perspective, the NPC Passenger either is or is not in violation of a restraining order, and at the beginning the player is simply unaware of whether the violation exists. However, we also aim to let the experience manager highlight the player's mistakes in a manner that is not simply due to chance. For instance, we would want to consistently discourage the player from preemptively arresting the Passenger based on an unsubstantiated guess that there is a restraining order violation, even if in practice the guess would happen to be correct some of the time. To show the player the possible consequences of a variety of mistakes, we have the experience manager model the underlying world as nondeterministic [22, 23]; e.g., rather than deciding at the beginning of the scenario whether the violation exists, the experience manager is free to invent the answer when it is needed, either by revealing whether there is a violation when the player looks up the Passenger in the database, or by deciding after an unjustified arrest that there was no violation and hence ensuring the player learns what could go wrong with their decision.

We explicitly model every possible world that is consistent with the player's current knowledge. We create problem instances for each possible setting of the nondeterministic variables, generate the deterministic story graph for each problem instance, and then track the current state in each of the story graphs in parallel. Each time the player observes an action in the simulation, we eliminate any story graph where the observation would not have been possible, and then update the remaining states to reflect the action. The available actions in the nondeterministic model consist of any action available in an individual member of the set. For instance, the experience manager may keep track of the state of two worlds, one in which the Driver has filed a restraining order against someone and one where there is no restraining order. If the player looks up the Driver's ID in the database, the experience manager may decide that a restraining order exists, forcing the other world to be dropped from consideration. This may eliminate future actions from being accessible, e.g., if there is no restraining order then the Passenger will not be motivated to harm the Driver. Alternatively, if the player never looks up the Driver's ID in the database and nothing else happens that would entail a definitive choice about the restraining order, the experience manager can choose between the two worlds at the very end of the scenario, or if it decides the Pas-

senger should harm the driver to convey its lesson.

3.6. Experience Manager Decision Making

The experience manager agent now has access to a nondeterministic story graph; at any given moment, the graph supplies the pruned set of all NPC actions that are explainable in some possible world consistent with the player's observations so far.

The purpose of the experience manager is to demonstrate the potential negative consequences of a player's actions. We consider best practice to be actions which make negative consequences less likely.

The experience management strategy that we evaluate in this paper tries to bring about the worst ending *that was enabled by a player action.* We do not simply try to cause bad endings whenever possible, which is trivial in this domain. For example, just as the story begins, the experience manager could decide that the Driver has a restraining order against the Passenger and that the Passenger is armed. The Passenger could then shoot the Driver at any arbitrary time, resulting in one of the worst possible endings without the chance for the player's decisions to change the outcome. This clearly would not serve the pedagogic goals of the simulation.

From some given state, we say an ending is available to the experience manager if there exists a sequence of explained actions that can be executed in that state, which results in that ending, and such that all actions require only NPC consent. When the player acts and causes a new ending to become available, the experience manager chooses NPC actions that drive the story toward the worst such ending. Suppose that the player acting as the Officer chooses to arrest the Passenger before they are certain that the Passenger is named in the restraining order. This action enables two endings: an innocent has been arrested (because the passenger is not actually named in the restraining order), or a criminal has been arrested (because they are named in the order). The experience manager responds to this by deciding that the Passenger was innocent. The experience manager makes decisions about domain facts instantaneously, but when pushing for an ending requires NPC interactions, these are rendered in real time, and the player may be able to prevent the worst ending that the experience manager attempts to cause.

4. Evaluation

Our hypothesis, broadly stated, is that our experience manager is more effective than a control at directing a playthrough trajectory to an ending appropriate for the player's decisions. In the future, we plan to integrate our experience manager with our virtual reality environment and validate our approach with human players. In the present preliminary study, we examine simulated playthroughs using the experience manager in isolation with a random player agent.

For both our experience management policy and the control, we generated the base story graphs for each combination of settings in our planning domain; applied lazy NPC, shorter plan, and dominant plan pruning; and generated the nondeterministic story graph from the pruned graphs. We then ran simulated playthroughs by starting at the root of the nondeterministic story graph and selecting actions until an ending was reached.

To select an action, we first chose randomly with equal probability whether the next action would be a player action or an NPC action. When the player was chosen to act, the player's action was chosen uniformly at random. When an NPC was chosen to act, the selection method depended on which policy was being used. For our experience management policy, we used the action selection strategy described in the previous section that guides the story toward player-enabled endings. For our control, we sampled NPC actions uniformly at random.

We ran simulated playthroughs in this manner until we had at least 200 occurrences of each ending with the experience manager and with the control, as some endings are more common than others. This gave us a total of 29,555 playthroughs with our experience manager and 34,002 playthroughs with the control.

We want to test the claim that the correlation between worse player decisions and worse endings in a playthrough is stronger with our experience manager than with the control. A ranking of "worse endings" is supplied by the planning domain's author utility function; however, we need a concrete definition of "worse player decisions". In future work with human subjects, we will examine ways to define the overall quality of a player's decisions in a playthrough. For the present work, we use occurrences of some specific player decisions as a proxy for this overall quality. In each simulated playthrough, we recorded the number of times the player made the following decisions:

(a) Checking the Driver's ID

(b) Learning whether any restraining order exists

(c) Checking the Passenger's ID when a restraining order exists

(d) Giving the Driver a traffic citation when there is not a restraining order against the Passenger

(e) Arresting the Passenger when there is a restraining order against them

(f) Checking the Passenger's ID when it has not been confirmed that a restraining order exists

(g) Giving the Driver a traffic citation when it has not been confirmed that there is no restraining order against the Passenger

(h) Arresting the Passenger when it has not been confirmed that there is a restraining order against them

(i) Shooting the Passenger when it has not been confirmed that there is a restraining order against them

(j) Arresting or shooting the Driver

(k) Giving away the Officer's gun

We considered (a) through (e) as best-practice decisions and (f) through (k) as contrary-to-best-practice decisions.

For each of these decisions, for our experience manager and for the control, we computed the Spearman correlation coefficient, choosen because it is suitable for ordinal data like the ending rankings, between the number of occurrences and the goodness of the ending given to the player. (Recall that the goodness of ending has five possible values. For the purpose of correlation, we assigned higher values to better endings.) We performed hypothesis tests for whether the correlations for our experience manager differed from the correlations for the control. We did the same analysis for the total occurrences of best-practice decisions, the total occurrences of all contrary-to-best-practice decisions, and the difference of the two totals.

Table 1 shows the correlation coefficients and the p-values for the hypothesis tests comparing them. The decisions are abbreviated in the table according to the list above.

A surprising result was that in almost all categories for both our experience manager and our control, there was a *positive* correlation between contrary-to-best-practice decisions and ranking of the resulting ending; that is, the more the player did the undesirable behavior, the better the ending the player tended to achieve. Note that the lone case with the negative correlation, (j), is arresting or shooting the Driver, which would immediately result in one of the worst endings of the simulation. The other contrary-to-best-practice decisions likely had positive correlation with ending quality because they tended to lead to the playthrough ending in a way that was suboptimal but did not involve assaulting the Driver, raising ending quality simply by averting the very worst endings. This reveals the limitation of a random player agent for evaluating our system; more structured artificial agents and eventually human players will be critical for later evaluations.

However, our results were encouraging overall because in all cases except (e) and (g), our experience manager outperformed the control. That is, our experience manager had a stronger correlation of good decisions with good endings and bad decisions with bad endings; a weaker correlation of bad decisions with good endings; or a weaker correlation of good minus bad decisions with bad endings. In five of these cases (p-values bolded in the table), the improvement was statistically significant ($p < 0.05$).

Table 1
Correlation between player decision frequencies and rank of the endings achieved in the sampled playthroughs. The lettered lines show correlations with individual decision types; the "good" and "bad" lines identify correlations with per-playthrough totals of all best-practice and contrary-to-best-practice decisions respectively.

decision	EM corr.	control corr.	p (EM $\neq$ control)
(a)	0.090	0.079	0.16
(b)	0.049	0.041	0.31
(c)	0.027	0.021	0.45
(d)	0.006	0.003	0.70
(e)	0.068	0.071	0.70
good	0.091	0.080	0.16
(f)	0.062	0.085	**<0.01**
(g)	0.020	0.014	0.14
(h)	0.460	0.610	**<0.01**
(i)	0.034	0.068	**<0.01**
(j)	-0.221	-0.182	**<0.01**
(k)	0.171	0.198	**<0.01**
bad	0.407	0.413	0.36
good−bad	-0.328	-0.338	0.20

5. Conclusions

Eventually, the playable simulation environment and the story-graph-based experience manager will communicate with each other to fully automate the human player's experience. An intermediate layer will observe how the player interacts with the environment and render those observations to the experience manager into high-level actions from the planning domain; conversely, the intermediate layer will take NPC actions selected by the experience manager and translate those actions into lower-level commands for the environment. Currently, however, both the simulation environment and the experience manager are in parallel development, and the environment relies on a human controller to send commands through a graphical interface. Future work on the experience manager component includes exploring ways to make the emotional planning model more scalable so we can generate longer stories, exploring more deeply the question of "What does it mean for the player to be responsible for an outcome?", and investigating how to model and guide the player's learning through multiple playthroughs.

Acknowledgments

We thank Officer Jennifer Melgar for consultation on our scenario design, Alireza Shirvani for implementation advice, and E.T. Garcia for work on animations.

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1911053. 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] A. Shirvani, S. G. Ware, A formalization of emotional planning for strong-story systems, in: AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2020, pp. 116–122.

[2] J. Robertson, R. M. Young, Perceptual experience management, IEEE Transactions on Games 11 (2018) 15–24.

[3] C. Martens, G. Smith, Towards a critical technical practice of narrative intelligence, in: Intelligent Narrative Technologies Workshop, 2020.

[4] W. Ma, O. O. Adesope, J. C. Nesbit, Q. Liu, Intelligent tutoring systems and learning outcomes: a meta-analysis, Journal of Educational Psychology 106 (2014) 901—-918.

[5] T. Bosse, C. Gerritsen, J. de Man, An intelligent system for aggression de-escalation training, in: European Conference on Artificial Intelligence, IOS Press, 2016, pp. 1805–1811.

[6] T. Bosse, C. Gerritsen, Towards serious gaming for communication training-a pilot study with police academy students, in: International Conference on Intelligent Technologies for Interactive Entertainment, 2016, pp. 13–22.

[7] M. O. Riedl, R. M. Young, Narrative planning: balancing plot and character, Journal of Artificial Intelligence Research 39 (2010) 217–268.

[8] S. G. Ware, R. M. Young, CPOCL: a narrative planner supporting conflict, in: AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2011, pp. 97–102.

[9] S. G. Ware, R. M. Young, Glaive: a state-space narrative planner supporting intentionality and conflict, in: AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2014, pp. 80–86.

[10] A. Shirvani, R. Farrell, S. G. Ware, Combining intentionality and belief: revisiting believable character plans, in: AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2018, pp. 222–228.

[11] Y. M. Cogollo, A. A. G. Salgado, L. A. M. García, Intelligent tutoring systems and planning techniques: A systematic review, Acta ScientiÆ InformaticÆ 4 (2020).

[12] J. Ramírez, A. De Antonio, Automated planning and replanning in an intelligent virtual environments for training, in: International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, 2007, pp. 765–772.

[13] N. Vannaprathip, P. Haddawy, H. Schultheis, S. Suebnukarn, P. Limsuvan, A. Intaraudom, N. Aiemlaor, C. Teemuenvai, A planning-based approach to generating tutorial dialog for teaching surgical decision making, in: International Conference on Intelligent Tutoring Systems, 2018, pp. 386–391.

[14] J. M. Thomas, R. M. Young, Annie: Automated generation of adaptive learner guidance for fun serious games, IEEE Transactions on Learning Technologies 3 (2010) 329–343.

[15] D. L. Roberts, C. L. Isbell, A survey and qualitative analysis of recent advances in drama management, International Transactions on Systems Science and Applications 4 (2008) 61–75.

[16] M. Nelson, M. Mateas, Search-based drama management in the interactive fiction *Anchorhead*, in: AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 1, 2005, pp. 99–104.

[17] E. T. Garcia, S. G. Ware, L. J. Baker, Measuring presence and performance in an intelligent virtual reality police use of force training simulation prototype, in: Florida Artificial Intelligence Research Society Conference, 2019, pp. 276–281.

[18] S. G. Ware, E. T. Garcia, M. Fisher, A. Shirvani, R. Farrell, Multi-agent narrative experience management as story graph pruning, IEEE Transactions on Games (2022).

[19] M. O. Riedl, R. M. Young, From linear story generation to branching story graphs, in: AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2005, pp. 111–116.

[20] S. G. Ware, C. Siler, Sabre: a narrative planner supporting intention and deep theory of mind, in: AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2021, pp. 99–106.

[21] R. E. Fikes, N. J. Nilsson, STRIPS: a new approach to the application of theorem proving to problem solving, Artificial Intelligence 2 (1972) 189–208.

[22] J. Robertson, R. M. Young, A model of superposed states, in: AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2016, pp. 65–71.

[23] J. Robertson, A. Amos-Binks, R. M. Young, Directing intentional superposition manipulation, in: AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2017, pp. 232–239.