Leveraging Intention Revision in Narrative Planning to Create Suspenseful Stories

Matthew William Fendt
Department of Computer Science
Baylor University
Waco, TX 76798

R. Michael Young
School of Computing
University of Utah
Salt Lake City, UT 84112

***Abstract*—Character intention revision is an essential component of stories, but it has yet to be incorporated into story generation systems. However, intentionality, one component of intention revision, has been explored in both narrative generation and logical formalisms. The IRIS system adopts the belief/desire/intention framework of intentionality from logical formalisms and combines it with preexisting concepts of intentionality in narrative. IRIS also introduces the crucial concept of intention revision for the protagonist of the story. The IRIS system uses its generative power alongside psychological and narrative models of suspense to computationally create suspenseful stories. It has been used in the creation of suspenseful non-interactive text stories and an interactive text-based video game.**

I. Introduction

The IRIS story generation system was designed as a tool for authors to aid in the creation of suspenseful stories. The result of the IRIS generation process is a suspenseful story outline that allows for some of the authorial burden of story creation to be offloaded to the system while still allowing the author to add his or her personal touch to the story. The IRIS system incorporates its definition of suspense from a successful existing generation system called Suspenser [1] into a generative process called planning, which is drawn from the field of Artificial Intelligence. The planning process is driven by the system's model of dramatic irony, which involves a disparity of knowledge between the audience and the protagonist of the story. As the protagonist struggles and reforms its plans in order to overcome the obstacles of the story, the audience feels invested in the outcome of the protagonist's actions. This method of story authoring was used in the successful creation of suspenseful narratives including non-interactive text stories and an interactive text adventure video game.

To this day, stories are still almost exclusively hand-authored. Systems that automatically or procedurally generate stories are in their infancy, largely due to the complications that creating interesting stories present. At best, the required structure of an interesting story is complex, and at worst, the structure is still unknown. Narratologists like Propp [2], who analyzed the components of Russian folk tales, have proposed common building blocks of narrative, but most problems in the field of story generation are still largely unsolved. One of the complications of computational story generation is the creation of believable portrayals of characters, their actions, and their intentions. One way to create these believable characters is to have them act in a coherent manner, performing only actions that stem from plausible motivations, updating their beliefs when appropriate, and replanning in the event of plan failure.

One way to model these believable characters is to represent their mental model using the cognitive belief/desire/intention (BDI) framework [3]. This captures 1) the character's beliefs, things it thinks are true about the world, 2) the character's desires, things it would like to make true in the world, and 3) the character's intentions: commitments to action. The BDI framework has many applications, such as representing speech acts [4], but when BDI is extended to include the concept of intention revision, it can be applied to narrative generation systems such as in the IRIS system.

The IRIS planning system is intended to address the issue of character intentionality and intention revision in the context of story generation. Intention revision is the modification of a character's plan or intentions due to a change in its beliefs. IRIS is a narrative planner that provides rules for adopting and dropping intentions in the context of narrative and uses a BDI mental model for the protagonist of the story.

One powerful way to leverage intention revision in a narrative generation system is to have the intention revision aid in the creation of suspense. Suspense is "the feeling of excitement or anxiety that audience members feel when they are waiting for something to happen and are uncertain about a significant outcome" [1]. Since IRIS's generation process provides a framework for instigating or preventing intention revision, IRIS has direct control over the manipulation of the individual characters' plans and goals, which it uses in conjunction with psychological and narrative models of suspense to create suspenseful stories.

To create this feeling of suspense in the audience, the IRIS system focuses on creating suspense around the protagonist of the story. IRIS's representation of the protagonist includes a robust BDI mental model that controls the protagonist's planning and intention revision. After the protagonist forms a plan, the generation process introduces threats to certain goals of the protagonist's plan using a set of suspense action templates. These suspense action templates were created for the purpose of the system and were experimentally validated. The suspense action templates create suspense at the story-action level, meaning suspense is created from the selection and ordering of the actions themselves. This is opposed to the discourse level, where suspense is created by the use of word selection, camera angles, lighting, and other authorial touches [5]. The IRIS system creates suspense by eliminating possible paths of success to achieving the target goal. The other, non protagonist characters, called supporting characters, are available to help perform these actions that are injected into the generation process. In the case of plan failure due to these threats, the protagonist performs intention revision to modify its plan and intentions. The process repeats until a complete suspenseful story is generated.

The intention of the IRIS system is to produce a story template that has intrinsic suspense at the story-action level and also has believable protagonist behavior. The output of the IRIS system can then be applied to both non-interactive and interactive contexts. Story fragments were chosen as a non-interactive medium and a text adventure video game was chosen as the interactive medium. The narratives in both of these domains were experimentally validated as effective at creating IRIS's intended form of suspense in the audience.

II. Contributions of the IRIS System

The contributions of the IRIS system are twofold. First, the system is a narratively driven planner that creates suspenseful story outlines. Second, the outlines it creates can be translated to a variety of different media, like non-interactive text and interactive text games.

In the field of narratology, there is a drive to study and formalize the fundamental narrative phenomena. One such phenomenon of interest is suspense. There are a number of ways to create suspense in a story. The IRIS system provides a concrete implementation of one such narratively-driven definition of suspense: "the feeling of excitement or anxiety that audience members feel when they are waiting for something to happen and are uncertain about a significant outcome" [1]. This kind of suspense is generated using the concept of dramatic irony, which is when the audience knows something that the characters do not. The audience will have knowledge of the protagonist's imminent failure and experience suspenseful anxiety as they watch the character work towards what they know to be a doomed plan and wonder how the protagonist will be able to succeed in its goals in the face of this impending failure. The IRIS system implements this definition of suspense through a model of protagonist plan failure and replanning.

The computational output of the IRIS system can also be translated into different media while preserving the sense of suspense from the initial outline. The ability for the outlines to be used in different media allows for an author to use the system to create his or her story outline, which can then be modified to allow for the author's personalization. The flexibility of the output can also allow for the automation of suspense creation in a game world. The outline created by IRIS system provides an alternative for a game designer to writing all of the suspenseful story moments ahead of time into the game. Additionally, the template nature of the output of the IRIS output allows for flexibility in its instantiation into the game world, allowing choice of how the suspense is realized in the game.

III. Related Work

Since narratives can provide significant cognitive benefits to the audience [6], [7], [8], researchers have been working over the last few decades to create narrative generation systems. Work in the 1970's and 1980's started with the study of knowledge structures and natural language [9], [10], [11], [12], [13], [14]. This work developed into some of the first computational narrative generators. These systems were for the most part heavily knowledge-based, meaning that they only functioned in very limited domains. Recently, the study of narrative and narrative generation has reached a broader interdisciplinary context, with tie-ins to art, psychology, culture, literary studies, and drama [6]. Games, too, have received attention for their narrative properties [15].

Some of these story generation systems have modeled suspense in the generation process. The MINSTREL system is a case-based reasoning system that creates a number of narrative phenomena including suspense. Suspenser is a planning system that also generates suspense, but unlike the IRIS system it creates suspense at the narrative's discourse level.

A. MINSTREL

MINSTREL [16] is a case-based reasoning system that creates stories that have, among other properties, suspense. It creates suspense mainly at the story-action level, as opposed to the discourse level. The original MINSTREL code is no longer available, though there has been an attempt at reimplementing it [17]. MINSTREL's method of creating suspense is derived from the example in the original publication. The only appropriate time to create suspense in its model is in a scene that the author wants to emphasize, and when the character's life is in danger. MINSTREL suspense formula is the following: 1) Make the target character sympathetic. 2) Choose an important scene. 3) Threaten that character's life. 4) Inject emotion at the discourse level. 5) Have the character try to get away and fail. MINSTREL is valuable because it provides a framework for when, when not, and how to create suspense around an outcome. Its weakness is the overly simplistic method of creating suspense, likely because many different narrative phenomena were present in the system and suspense was not the sole focus of the work.

B. Suspenser

Suspenser [1] is a planning system that generates suspenseful stories. It generates suspense at the discourse level. Suspenser first uses a planner to produce the story events. It then iteratively increases suspense by replacing and reordering story events to create a discourse. Suspenser is the closest system to IRIS in that they are both planning systems that focus on creating suspenseful stories. However, since Suspenser generates suspense at the discourse level, it is not generating stories that are intrinsically suspenseful, but rather adding suspense to the stories after they have been generated. Suspenser also is not guaranteed to generate high suspense stories, since it can get stuck in a local maximum when adding suspense.

IV. IRIS Story Generation Algorithm

A. Overview

The Intention Revision in Storytelling (IRIS) system is a story generation system designed to create suspenseful story outlines. An overview of the system is shown in Figure 1. The outer orange box represents the system as a whole. The user of the system provides IRIS the initial story domain information, such as actions that can happen in the story and a list of characters.

This information is used by the system's planner, which is represented by the blue box. A planner is a tool that allows for action selection and sequencing, and when applied to story generation, story action selection and ordering. The IRIS planner creates the story plan around the protagonist, who is at the focal point of the story. To allow for the protagonist to behave in a believable and coherent manner, the protagonist is provided with two psychological properties. The first is a Belief/Desire/Intention (BDI) framework [3], which allows for the protagonist to have its knowledge and beliefs segmented from the rest of the world. The second is a model of intention revision, which dictates how the protagonist will behave in the face of failure throughout the story. The protagonist of a story frequently encounters difficulties, and intention revision provides a systematic approach through which the character struggles to accomplish its goals. Both the BDI framework and intention revision contribute to the creation of suspense and are manipulated by the introduction of suspenseful action sequences, which are shown in the green box.

The IRIS planner contains a set of experimentally validated suspenseful action templates that it uses in the planning process. These templates can be translated into action sequences which are introduced to create the suspense in the story. Suspense is created by an antagonistic character taking an action which will thwart the protagonist's goal. This malicious action, which is known to the audience but unknown to the character, is afforded by the use of the BDI mental modeling discussed above. The audience feels suspense as the character unknowingly proceeds with its doomed plan. The suspense comes to a climax when the character realizes its failure and must perform intention revision to find another way to achieve its goal.

The output of the IRIS system is a story outline with suspenseful story moments, as represented by the dark blue box. Due to the overhead that narrative planning introduces, the author needs to initially populate the IRIS system with information about the story world, such as the characters and possible actions. However, the payoff is that the IRIS planning process can produce a number of story outlines in the given domain which can be instantiated into different media. In particular, the IRIS system has been shown to be effective at creating suspenseful, non-interactive text story fragments

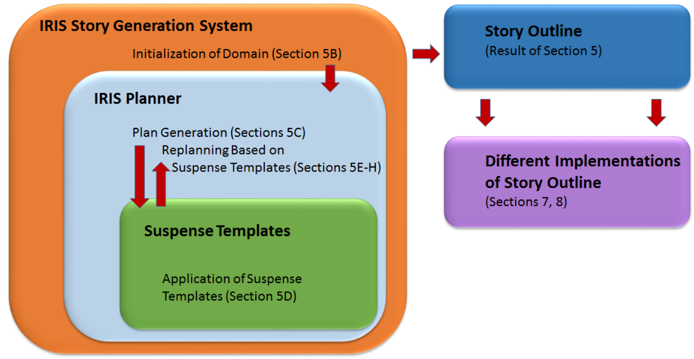

Fig. 1: The IRIS story generation system. The generation process creates story outlines containing suspense that can then be instantiated in different media.

Algorithm 1: IRIS-Narrative-Generation Algorithm(C, G_A, Λ)
Initialization: Let $P = \emptyset$. For the protagonist p, I_p = create-intentions(D_p, G_A). $G_S = G_A \cup \forall i \, \epsilon I_p$ such that i is a high value intention. $g_c = \emptyset$.
1. **Protagonist Plan Creation:** If $P_p = \emptyset$, P_p = find-plan(B_p, I_p, Λ). g_c is the temporally first g ϵ G_S.
2. **Application of Suspense Templates:** If g_c does not have a suspense template applied to it, identify the actions in P_p that accomplish g_c. Choose one of these actions to threaten. Apply an applicable suspense template. Fill in the minimum number of variables in the template with consistent terms.
3. **Drama Manager Action Addition:** If next Drama Manager Action a is not a ground term (free of variables), fill in variables consistent with world conditions and character intentions. Add a to P. If an effect of a is g_c, remove g_c from G_S and $g_c = \emptyset$.
4. **Belief Update:** Perform belief update based on the Drama Manager action.
5. **Intention Revision**: Call Intention-Revision(p, a)
6. **Repair of Suspense Templates:** Update g_c's template with consistent terms.
7. **Recursive invocation**: If $G_S \neq \emptyset$, call IRIS-Narrative-Generation(C, G_S,Λ).

Fig. 2: Algorithm 1: IRIS-Narrative-Generation Algorithm

[18] and a suspenseful, interactive text adventure video game [19]. The IRIS system is a useful tool for authors because it can provide a suspenseful basis of a story to which the author can then add his or her own personalization.

V. IRIS Generation Algorithm

A. Overview

Figure 2 describes IRIS's narrative generation algorithm. The input for the algorithm is a set of characters C containing a protagonist character with initial BDI mental model (initial beliefs and weighted desires) and a list of supporting characters with optional 'friend' or 'foe' labels, a set of authorial goals G_A, initial world conditions, and a set of actions Λ. This initialization of the planning domain will determine the genre and nature of the story. The output is a totally ordered complete plan where all of the authorial goals are satisfied.

The narrative generator first takes in this initial information about the story. The protagonist character has a set of beliefs and desires. The supporting characters do not have this BDI model, but rather serve to enhance the suspense created around the protagonist. The authorial goals are assigned to the protagonist and the protagonist forms its maximally-weighted internally consistent subset of intentions from its desires and assigned authorial goals. Authorial goals are different than character goals [20] in that they are imposed upon the protagonist by the author and have higher priority than character goals if there is a conflict between the two. The protagonist now has a set of beliefs as a starting state and intentions for an ending state, so it uses these to form its initial plan.

A Drama Manager [21], [22] is used to monitor the story and finds the next goal around which to create suspense. It applies suspense templates that threaten the completion of that goal. The suspense templates are modular templates designed to create suspense around the completion of a goal (See Section V-D). Then, the Drama Manager adds the next appropriate Drama Manager action to the story. The protagonist has the opportunity to update its beliefs and perform intention revision based on the effect of this action. Any Drama Manager actions added by the application of the suspense templates that were damaged by the selection of the last Drama Manager action are now repaired. Finally, if there are no more suspenseful goals to satisfy and the system is not in the middle of fulfilling a suspenseful goal, the algorithm returns the completed plan. Otherwise, the algorithm is called recursively.

B. IRIS's Belief/Desire/Intention Framework (Initialization)

The protagonist may initially have a set of character desires, which are conditions that the protagonist has not yet committed to acting upon. The authorial goals are added to this set of character goals. An internally consistent subset of conditions need to be found to be converted into protagonist goals. This is done by first prioritizing the authorial goals, with as many character desires added in as can be made consistent with the set of authorial goals. This set of conditions is now labeled as the protagonist's goals, which the protagonist will now form a plan to complete. The set of suspenseful goals G_S is set to these protagonist goals.

C. Protagonist Plan Creation (Step 1)

If the protagonist's plan is null, the protagonist creates a plan by applying its BDI mental model to planning. The beliefs serve as a start state, and its intentions as the goal state. The current suspenseful goal g_c is set to the first goal in G_S to be achieved by the protagonist's plan. IRIS uses the partial order planning (POP) algorithm called Longbow [23]. The entirety of the IRIS story generation is all done off-line and not in real time. There is not a sense of mixing planning and execution. Even the intention revision that takes place over the course of story generation is still done at planning time. The intention revision occurs during story generation when the character can no longer adhere to its plan and/or intentions that it held previously in the generation process.

D. Application of Suspense Templates (Step 2)

IRIS creates suspense from protagonist plan failure. The protagonist's plan will fail if one of the preconditions of the actions in its plan does not hold. In this step, a threat by an antagonistic character is injected into the protagonist's plan. There are three ways that the precondition can fail to hold.

1) Conflicting Action `a`, Withhold True Observation of `a`, Failed Action `b`, True Observation of `b`
2) Conflicting Action `a`, Character not present for True Observation of `a`, Failed Action `b`, True Observation of `b`
3) True Utterance `a`, Conflicting Action `b`, Withhold True Observation of `b`, Failed Action `g`, True Observation of `g`
4) True Observation `a`, Conflicting Action `b`, Withhold True Observation of `b`, Failed Action `g`, True Observation of `g`
5) True Utterance `a`, Conflicting Action `b`, Character not present for True Observation of `b`, Failed Action `g`, True Observation of `g`
6) True Observation `a`, Conflicting Action `b`, Character not present for True Observation of `b`, Failed Action `g`, True Observation of `g`

Fig. 3: List of the validated IRIS suspense templates. These templates consist of a series of Drama Manager actions, that when inserted into the protagonist's plan before a given protagonist goal, create dramatic irony and suspense around the goal.

The first way a precondition can fail to hold is an inconsistency between the protagonist's beliefs and the actual state of the world. This inconsistency can be unknown to the audience too, so that the audience is surprised by the plan failure, or known to the audience, resulting in dramatic irony.

The second way a precondition can fail to hold is if the system deliberately modifies the state of the world to introduce plan failure. In this case, a precondition of the protagonist's plan was true up to a point, but then was reversed by the system. This is analogous to initial state revision in planning [24]. This modification is unknown to the protagonist, but also unknown to the audience. It is kept secret from the audience because the change is a retroactive modification to the initial state of the world which seemingly has no perpetrator, so it would look unnatural for the audience if they saw it changed.

The third way a precondition can fail to hold is if an antagonistic agent takes action to thwart a precondition in the protagonist's plan. This agent could be another character or natural forces. The action may or may not be observed by the protagonist. It is observed by the audience, because otherwise this condition would be a subset of the first condition where, unknown to the protagonist and the audience, the state of the world is inconsistent with the protagonist's beliefs.

IRIS's suspense templates were created from a manual enumeration of all of the Drama Manager actions that resulted in these kinds of protagonist plan failure. Drama Manager actions are discussed in Section V-E. The suspense templates listed in Figure 3 were experimentally validated as being suspenseful (See Section VII) and integrated into the IRIS story generation algorithm.

In this step of the algorithm, if g_c, the currently targeted suspenseful goal, does not have a suspense template assigned to it, then one needs to be assigned. A suspense template is applied to a goal as follows:

1) Identify the actions that are part of the plan to achieve g_c. This can be done by using a simple algorithm that runs in $O(n)$ that works backwards from the end of the plan and adds actions to a list that are needed to satisfy the preconditions of the final action or another action added to the list in this way. This is a similar process to causal chaining [25].

2) Choose one of these actions non-deterministically, selecting from an action labeled by the author as conducive to suspense if possible.

3) Select a precondition of the chosen action to invalidate. To do this, the system first non-deterministically selects an antagonistic character. Then, for each precondition of the chosen action, the system tries to find a plan for the antagonistic character with the current state of the world as the start state and the negation of each precondition in turn as the goal state. Those preconditions without a resulting plan are eliminated from consideration.

4) For every precondition with an identifiable antagonistic plan, the system then sees if the protagonist could perform intention revision to reestablish the invalidated precondition. Those preconditions that the protagonist cannot reestablish are eliminated from consideration.

5) A precondition is selected non-deterministically from the remaining preconditions.

6) The antagonistic actions that invalidate this precondition need to be incorporated into a suspense template (see Figure 3). One of the applicable templates is chosen non-deterministically. This will be used to create a suspenseful moment in the story. A future extension to IRIS might provide a relative ranking of the templates with regards to suspense for a given situation. Bindings in the template are filled in with a least-commitment approach.

7) The suspense template actions are inserted into the plan directly before the action that completes the suspenseful goal.

For an example of the application of a suspense template, see Section VI.

E. Drama Manager Action Addition (Step 3)

The Drama Manger adds the next action from the mutable protagonist plan to the immutable story plan. If the next action contains ungrounded variables (meaning it is an action from a suspense template), the variables need to be grounded at this time.

The Drama Manager has the following available actions:

Select Character Action(A, L)
Description: "The Drama Manager selects an action `A` from a character's plan. The action happens at location `L`"

For this action, the Drama Manager simply selects an action from a character's plan. In the way that IRIS is defined, only the singular protagonist will have a plan, so the protagonist will always be the one whose plan is selected. This action is left in its general form if the system is ever extended to allow for multiple protagonists.

After the effects of the actions are updated in the world conditions, the protagonist observes all of the effects of the action as well.

True Observation(X, N, L)
Description: "`X` correctly observes `N` at location `L`"

For this action, the protagonist observes some true state of the world that is observable from its location. The protagonist then adopts this world condition in its beliefs.

True Utterance(X, Y, N, L)
Description: "`X` truthfully informs `Y` of `N` at location `L`"

For this action, one character conveys a belief that it holds to the protagonist that is present at the same location. The protagonist then adopts this world condition in its beliefs.

Conflicting Action (X, A, L)
Description: "An antagonistic character `X` takes an action `A` at location `L` that will thwart a precondition of a step in the protagonist's plan."

Here, the antagonistic character is introducing a threat to a precondition of the protagonist's plan in order to help create suspense.

F. Belief Update (Step 4)

If a `Select Character Action` was chosen in step 3, then the protagonist's beliefs are updated with the effects of the action that was selected.

G. Intention Revision (Step 5)

Based on the effects of the selected action, the protagonist might need to perform intention revision (See Figure 4).

Intention revision is the addition or subtraction of a character's intentions and/or the modification of the character's plan. Intention revision is not to be confused with plan revision. Plan

Algorithm 2: Intention-Revision (c, v)
1. **Working Assumptions Violation Check**: See if any of the effects of the chosen action `a` violate the working assumption of the character `c`.
2a. **Optimization Prune**: If `c` did not have a working assumption violation, remove each action `a'` that is an element in `c`'s plan p, such that $\forall$`e` $\in$ `a'`, `e` is an effect of `a'`, `e` $\in$ ({All world conditions} - {the preconditions of all subsequent actions in p}) || (the belief update from the effects of selected action `a`). Return.
2b. **Replan With New Beliefs:** If `c` did have a working assumption violation, P_c = find-plan(`c`, B_c, I_c, Λ)
3. **Drop Intentions Until a Working Plan Can be Found**:If $P_c = \emptyset$, P_c = find-plan(`c`, B_c, I_c, Λ). If P_c still $= \emptyset$, I_c = find-max-consistent-subset(I_c), then repeat this step.

Fig. 4: Algorithm 2: Intention-Revision Algorithm

revision is only an alteration of the actions in a character's plan, without considering its intentions. Plan revision is an aspect of intention revision, though. Intention revision consists of:

Working Assumptions Violation Check: See if any of the effects of the selected action violate the working assumption of the protagonist. Working assumptions are the minimal set of world conditions that, when violated, will prompt intention revision. At a given time `t`, the working assumptions consist of the preconditions of the actions of the unexecuted plan that are not satisfied in an effect of an action in the remaining plan. An example of a character's working assumptions is the following:
Given a character with the following actions and the actions' preconditions and effects:

```
Action 1:
  Preconditions: A, B   Effects: C, D
Action 2:
  Preconditions: C, E   Effects: F, G
Action 3:
  Preconditions: A, H   Effects: I, J
```

The character's working assumptions are `A, B, E, H`.

Optimization Prune: If the protagonist did not have a working assumption violation, then it does not need to replan or modify its intentions. However, it is possible that some of the actions in its plan are now unnecessary due to the effects of the selected action. For example, perhaps the character planned to sneak up to the guard, steal his keys, unlock the door, and walk through. The selected action caused an explosion that destroyed the door. It is now unnecessary to perform the first three steps in the character's plan, since it can now just walk through the door. The first three steps should be culled. This is done by removing all actions in the character's plan whose only effects are already satisfied or were just satisfied by the effects of the action. After this optimization prune, go to step 6 of Algorithm 1.

Replan With New Beliefs: If the protagonist did have a working assumption violation, then it needs to repair its plan, intentions, or both. First, try to replan with the same intentions as the goal state and its updated set of beliefs as the start state.

Drop Intentions Until a Working Plan Can be Found: If replanning alone did not produce a plan to satisfy all of its intentions, some intentions need to be dropped. A minimally weighted, inconsistent subset of intentions is dropped until a plan can be found.

H. Repair of Suspense Templates (Step 6)

It is possible that the effects of the chosen action made it such that the actions of the suspense template are now inconsistent with the world conditions (for example, if a character is expected in one of these actions but is now dead). If this is the case, repair these actions by choosing new ground terms consistent with the world conditions.

I. Recursive invocation (Step 7)

If there are still more suspenseful goals unsatisfied, call the algorithm recursively.

VI. Example Story Generation

This is an example portion of story generation that the IRIS system produced in its Western heist domain. The current suspenseful goal is `(has PLAYER DYNAMITE)`. The protagonist's current plan is shown in Figure 5.

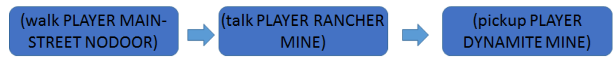

Fig. 5: The current protagonist plan before the suspenseful action sequence is inserted.

A suspense template needs to be applied to this goal. The system first finds the actions that contribute to the goal of `(has PLAYER DYNAMITE)`. In this case, it is both `(walk PLAYER MAIN-STREET MINE NODOOR)` and `(pickup PLAYER DYNAMITE MINE)`. `(talk PLAYER RANCHER MINE)` is an action that satisfies another goal, so it is not considered. Next, the system selects one of these actions to threaten. Since in the Western domain the pickup action is labeled as suspense-conducive and walk is not, pickup is selected. If there were multiple suspense-conducive choices, then one would have been selected non-deterministically.

The system now needs to select a precondition of the pickup action from the protagonist's plan to invalidate. The `(pickup PLAYER DYNAMITE MINE)` action and its five preconditions are shown in Figure 6.

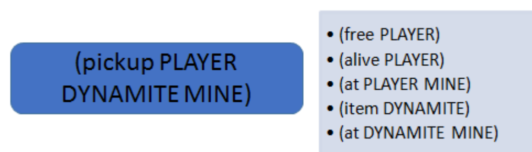

Fig. 6: The preconditions for the suspense-conducive pickup action. An antagonistic character will take a series of actions that threatens the completion of one of these preconditions.

Next, the system chooses an antagonistic character to take a series of actions that will invalidate one of these preconditions. The `MINER` character is selected non-deterministically from the available choices. Given the current state of the world as the start state and each of the preconditions in turn as the end state, the system tries to find a plan for the miner that invalidates the given precondition. In this example, there is no set of actions that can result in `(not (item DYNAMITE))`, so that precondition is eliminated from consideration. `(not (at PLAYER MINE))` is already true since the player is currently at the main street, so this condition already holds and is eliminated. The remaining three preconditions do have associated plans that can be found, as shown in Figure 7.

The system then sees if the protagonist, through intention revision, would be able to achieve the suspenseful goal of `(has PLAYER DYNAMITE)`. This is because the system wants to pose challenges to the protagonist, but ultimately have the protagonist succeed in its goals. If the `(alive PLAYER)` precondition is invalidated, the protagonist will not be able to perform intention revision to

Fig. 7: Antagonistic plans that can be generated to invalidate the different preconditions of the suspenseful action.

fulfill the `(has PLAYER DYNAMITE)`. Therefore `(not (alive PLAYER))` is eliminated from consideration. For the other two preconditions, Figure 8 shows the plans that the protagonist could adopt to complete the suspenseful goal.

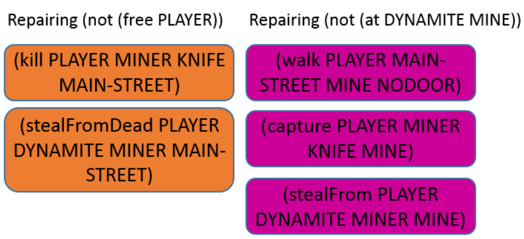

Fig. 8: Different plans the protagonist can adopt to repair the violated preconditions of the suspenseful actions.

Therefore, both of these preconditions are eligible to be invalidated by an antagonistic character's plan and then subsequently repaired. One precondition is selected non-deterministically, and for this example we will say `(not (at DYNAMITE MINE))` is selected.

The conflicting action sequence now needs to be integrated into a suspense template. The subset of available suspense templates that apply are:

1) Conflicting Action(pickup MINER DYNAMITE MINE)
(withhold True Observation of (has MINER DYNAMITE))
Failed Action((pickup PLAYER DYNAMITE MINE))
True Observation(has (MINER DYNAMITE))

2) True Utterance(?FRIENDLY-CHARACTER,
(at DYNAMITE MINE)))
Conflicting Action((pickup MINER DYNAMITE MINE))
(withhold True Observation of (has MINER DYNAMITE))
Failed Action(pickup PLAYER DYNAMITE MINE)
True Observation(has (MINER DYNAMITE))

One of these is selected non-deterministically, say the first. These actions are inserted into the plan immediately before the `(pickup PLAYER DYNAMITE MINE)` action. The actions currently in the plan are shown in Figure 9.

The first four actions in the plan are selected one at a time, removed from the plan, and added to the story. After the `Failed Action (pickup PLAYER DYNAMITE MINE)`, the protagonist performs an intention revision and realizes that the `(at DYNAMITE MINE)` condition for its

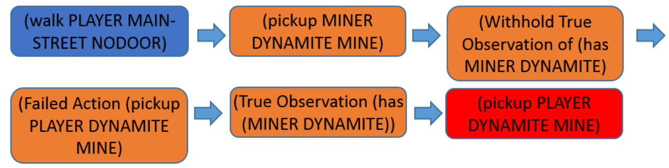

Fig. 9: The current story plan after the suspenseful actions sequence is inserted. The orange boxes indicate the actions that constitute the suspenseful action sequence that was inserted into the protagonist's plan. The red box represents an action originally in the protagonist's plan that will now fail because of the suspenseful action sequence.

action did not hold. It also notices that it can retain the `(has PLAYER DYNAMITE)` goal if it modifies its plan. This was anticipated earlier when the `(at DYNAMITE MINE)` precondition was being selected to be threatened. The character replans and the plan is shown in Figure 10.

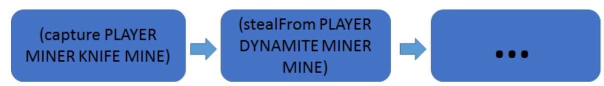

Fig. 10: The protagonist plan after the character performs intention revision due to a failed action.

These actions are successfully executed and the suspenseful goal of `(has PLAYER DYNAMITE)` is completed. The process now continues for the next suspenseful goal.

The complete set of story actions for this suspenseful goal is shown in Figure 11.

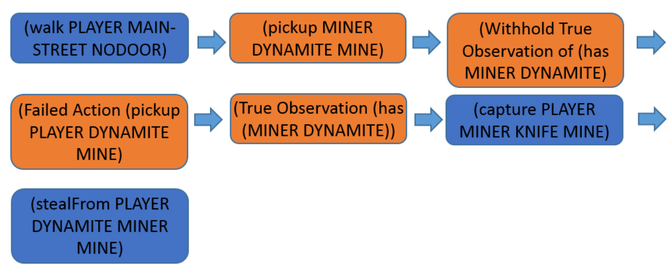

Fig. 11: The complete set of actions for the suspenseful goal.

A textual description of the events might be "The player wants to get the dynamite from the mine. However, the miner who is in the mine grabs the dynamite. The player moves to the mine, searches for the dynamite and can't find it. The player then realizes that the miner has the dynamite, threatens the miner with a knife, and steals it from the miner."

VII. Evaluation of IRIS in the Context of Non-Interactive Text Stories

The IRIS system was evaluated to be effective at creating story-level suspense in the domain of non-interactive text stories [18]. In the evaluation, four different suspense creation processes were compared: the IRIS suspense templates (discussed in Section V-D), an operationalization of a case-based reasoning system with narrative properties called MINSTREL [16], a human author with creative writing experience, and a logically consistent random baseline. Participants evaluated pair-wise comparisons of story fragments created by these different processes with regards to suspense. The story fragments produced by the human author and by most of the IRIS suspense templates, particularly the templates involving a disparity of knowledge between the audience and the protagonist, were significantly more suspenseful than the random story fragments. Additionally, the IRIS suspense templates did not perform significantly differently than the human author. This shows IRIS's effectiveness at creating story-level suspense in a reader in the domain of a non-interactive text story.

A. Experimental Design

All of the different creation methods being tested produced story fragments using the same fragment creation grammar. The domain was a murder mystery. In this story, there is a detective that is trying to figure out who killed the host of a dinner party. This is the goal around which the story fragments tried to create suspense.

To prevent the possible introduction of discourse-level suspense, a grammar was used for the creation of story fragments. The grammar is discussed in detail in a previous paper [18]. The grammar was very template-based in nature and was designed to minimize wording that would introduce discourse-level suspense. An example production rule of the grammar is *<person> moves to <place>*. Fragment length was limited to range between 2 and 6 sentences in order to prevent readers from potential distraction regarding the suspenseful goals. Additionally, the same person or place was not allowed to be used for both parts of a fragment. For example "The maid attacks the maid" was not allowed.

A human author with creative writing experience was recruited to produce suspenseful story fragments. He was given the creation grammar, the rules described in the previous paragraph, and told to create as many suspenseful story fragments around the goal of the detective trying to find the killer as he felt he could generate. He produced seven story fragments between lengths four and six sentences.

Next, a set of story fragments were created using the MINSTREL method of creating suspense (see Related Work section). The original MINSTREL code is no longer available, so the formula for creating suspense was derived from the original publication. In order to avoid the introduction of discourse level suspense, the step in MINSTREL where emotion is interjected at the discourse level was omitted. An example MINSTREL fragment was "The maid attacked the detective. The detective tried to move to the attic but failed."

Then, story fragments from the IRIS suspense formulas were created. There were multiple ways to create fragments from each of the suspense formulae, so all of the combinations were generated.

For the logically and temporally consistent baseline, a program was created that generated random story fragments following the story creation rules. The actual story fragments were generated at run time during each participant's experimental session.

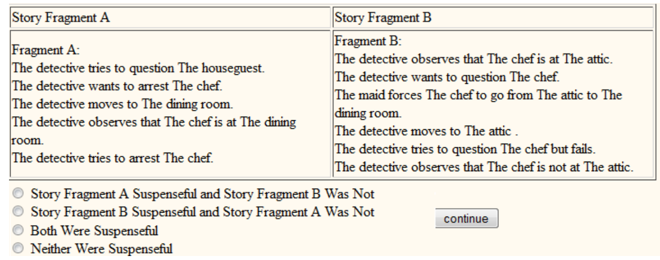

Fig. 12: A screenshot of the survey, where participants read and compared the suspensefulness of two story fragments.

The study was conducted online in HTML with JavaScript to help handle the selection and display of the story fragments. Participants were recruited by snowball sampling in email and in person, and through undergraduate and graduate computer science classes. For students recruited through courses, participation fulfilled an experimental course requirement. The participants were first given a link to the consent form. Then, they were given a demographic survey asking their age and gender. 83 people completed the demographic survey and were presented with the experiment, with an average age of 22.1 +/- 2.7. In the demographic survey and the comparisons, the participants always had the option to leave any question unanswered. Next, they were given the background information about the domain of the story.

The participants were then given a series of sentence fragment pairs. They were asked to read each fragment and indicate which was more suspenseful. They were given four options: Story Fragment A was suspenseful and Story Fragment B was not, Story Fragment B was suspenseful and Story Fragment A was not, Both were suspenseful, or Neither were suspenseful. After selecting their choice and hitting the Next button, they were shown another pairing. A screenshot of the survey page is shown in Figure 12.

Forced-choice pairwise comparison was chosen instead of a rating-based Likert scale for comparing the story fragments generated by the different suspense creation methods. This is because the suspense creation methods are being compared in their ability or inability to create suspense, and not how much more suspenseful one story fragment is than another. Additionally, forced-choice helps prevent some of the difficulties that a rating-based system can introduce when comparing across treatments, like the fact that the meaning of each level on a rating scale may differ across experiment participants and the assumption that equal ratings are considered equal value [26].

There were 36 pair-wise combinations of suspense creation methods: 11 IRIS action templates x 3 other story creation methods + the 3 combinations of these 3 methods. At run time, a random fragment from the possible ones of the given approach was selected.

Four hypotheses were tested:

Hypothesis 1: The story fragments generated with the IRIS suspense templates will be no different in terms

of suspense than the fragments generated by the human authors selecting from the same set of story actions. To ensure that the story fragments that the suspense templates and the human authors generate are able to be compared, they both compose their fragments from the same set of story actions. This puts a limitation on the human author since he or she can only use a limited set of story actions instead of an unlimited vocabulary. If this hypothesis holds, then there will be no evidence to suggest that humans produce more suspenseful stories than IRIS when selecting from the same set of story actions.

Hypothesis 2: The story fragments generated with the IRIS suspense templates will be more suspenseful than the randomly-generated baseline. It needs to be shown that the suspense templates are an improvement over no method of creating suspense at all.

Hypothesis 3: The story fragments generated by the human authors will be more suspenseful than the randomly-generated baseline. This demonstrates that the human authors were actually able to create more suspense than random combinations of story events.

Hypothesis 4: The story fragments generated with the IRIS suspense templates will be at least as suspenseful as the MINSTREL formula. Since the IRIS suspense templates are more complicated to implement and apply than the MINSTREL formula, it is desirable that they perform at least as well as the simpler MINSTREL approach.

B. Results

To evaluate if the IRIS templates would be rated comparably to the human authors and if the templates would be rated comparably to MINSTREL, the two-tailed 2x2 Chi-square test without Yates' correction was used to compare the participants' response to a null hypothesis of equal responses for both approaches. For each pair of story fragments, the participants could indicate that one, both, or neither of the story fragments were successful (See Figure 12). For example, of the people who were shown the story fragments pairing generated using template G and MINSTREL, 32 people reported that only the fragment from template G was suspenseful, 7 people reported that only the fragment from MINSTREL was suspenseful, 12 reported them both suspenseful, and 12 reported neither suspenseful. The participants did not compare the two story fragments to the other when determining whether each was suspenseful.

In this example, the 12 that reported both were suspenseful were added to the votes for each of the two approaches, bringing the count up to 44 votes for template G and 19 for MINSTREL. This is a total of 63 suspense votes. If the two techniques were comparably suspenseful, then instead of one approach getting 44 votes and the other getting 19, each would have received about half (31) of the votes. This is where the Chi-square test was applied. In this case, $\chi^2(1) = 5.13$, $p = 0.02$, indicating that the participants report template G more suspenseful than MINSTREL.

To evaluate Hypothesis 2, that the IRIS templates would be rated higher than random, and Hypothesis 3, that the human authors would be rated higher than random, we used the one-tailed 2x2 Chi-square test without Yates' correction to compare the participants' response to a null hypothesis of equal responses for both approaches. Here the process was similar to before, except that the responses "Both were suspenseful" were discarded since we wanted to measure if one performed higher, not just comparable.

In all cases, if more than 1/3 of the responses were "Neither were suspenseful," then the above tests were not performed. This is a threshold we decided on to indicate that the participants found both approaches to lack suspense.

One-tailed 2x2 Chi-Squared tests were performed to measure if MINSTREL and the human author performed significantly better than random. The test for MINSTREL failed to reject the null hypothesis of performing comparably to the random base line, ($\chi^2(1) = 0.01$, $p = 0.45$), but the test for the human author rejects the null hypothesis of performing comparably to the random, ($\chi^2(1) = 2.33$, $p = 0.06$).

Tables I, II, and Figure 13 show the experiment's results.

TABLE I: p values for IRIS suspense templates versus the three other creation processes in the experiment. A one-tailed Chi-Squared test was performed versus the random baseline, and a two-tailed Chi-Squared test was performed versus the other two. A dash indicates that more than 1/3 of the responses rated neither as suspenseful. An asterisk indicates that if a two-tailed test would have been performed, the p value would have been outside the p=0.05 interval.

	Random	MINSTREL	Human
Template A	-	0.46	0.06
Template B	-	-	**0.0001**
Template C	**0.001**	**0.01**	0.47
Template D	**0.003**	**0.05***	0.92
Template E	0.35	0.44	0.12
Template F	0.34	0.52	0.09
Template G	**0.002**	**0.02**	0.10
Template H	**0.001**	**0.05***	0.39
Template I	0.13	**0.04***	0.52
Template J	**0.04***	0.17	-
Template K	-	-	0.51

TABLE II: Table of counts for the participants voting for the suspensefulness of competing story fragments in each pairwise comparison. The number in bold is the count for the successful templates that were implemented into the IRIS system as a result of this experiment (See Section VII.C) and the number in parenthesis is the total count with all templates, both successful and unsuccessful.

	Templates v Random	Templates v MINSTREL	Templates v Human
Vote For Templates	**161** (222)	**169** (253)	**102** (138)
Vote For Other	**46** (113)	**36** (106)	**66** (195)
Vote For Both	**48** (82)	**73** (105)	**77** (135)
Vote For Neither	**73** (212)	**62** (153)	**63** (142)

C. Discussion

Hypothesis 3, that the human author would perform better than random, was supported. This is important because a human author is often considered the ideal that story generation

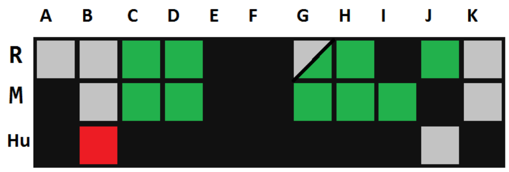

Fig. 13: A visualization of the comparison between the IRIS suspense templates and the three other creation processes in the experiment. Red indicates the the column was rated higher, green that the row was rated higher, black that they were not significantly different, all at p = 0.05 level. Gray indicates that more than 1/3 of the responses rated neither as suspenseful. One comparison was right at this 1/3 but had a highly significant p value.

systems strive towards, and it needs to be shown that the human author performed well enough to make useful claims when comparing a generation system to the human author.

Hypothesis 4, that the IRIS templates would perform at least as well as the operationalization of MINSTREL, was also supported. The IRIS fragments performed much better than MINSTREL, and MINSTREL did not perform significantly differently than random. This could be due to the fact that MINSTREL requires some discourse level suspense that was purposively minimized in the experiment. This is not an exhaustive comparison to MINSTREL since the MINSTREL program was not run, but rather an operationalization of its suspense-producing algorithm presented in the MINSTREL paper. Further work would include using a reimplementation of MINSTREL like MINSTREL Remixed [17] and doing a more detailed comparison of IRIS and MINSTREL.

For six of the 11 IRIS templates, Hypotheses 1 and 2 were supported as well. The exceptions were templates A, B, E, F, and K. These templates were removed from the IRIS generation algorithm. The most successful templates were those where the audience had knowledge about the protagonist's imminent plan failure and the protagonist did not. This makes intuitive sense and matches the system's definition of suspense which involves anxiety and uncertainty around the success of an outcome from the audience's perspective. These results show that there is no evidence to suggest that a human author selecting from the same set of story actions as the IRIS system using a small set of action templates produces more suspenseful story-action level story fragments.

VIII. Evaluation of IRIS in the Context of an Interactive Text Adventure Game

The IRIS system was also evaluated to be effective at creating suspense in the domain of an interactive text adventure video game [19]. One complication of using the IRIS output in such a system is that the audience and the protagonist are the same person. Therefore, it is not possible to create dramatic irony using a disparity of knowledge between the audience and the protagonist. However, it was hypothesized that IRIS's existing method of creating suspense, which also involves plan failure, replanning, incomplete knowledge, and an antagonistic force, could be adapted to an interactive context as long as two restrictions were placed on the game domain to which the IRIS output is applied. These restrictions are that players need to be informed of ways to replan towards their goals in the case of plan failure and limitations need to be placed on player actions ensuring that they do not bypass suspenseful story moments. The chosen method of implementing these restrictions was 1) an in-game method of giving hints of how to achieve the game's goals and 2) the presence of the suspenseful action sequences from the original story outline and the restriction of player agency during their execution. An interactive text adventure video game was created using the IRIS output with these restrictions on the game domain and evaluated to measure the suspense the game creates in its players.

In this experiment, participants played one of four versions of a text adventure video game in which one, both, or neither of the restrictions were used. After the game, participants answered survey questions that were create from the operationalization of IRIS's definition of suspense. These questions measured participants' sense of suspense during the execution of the three suspenseful goals from the story outline and several other non-essential, non-suspenseful goals. It was found that the version of the game containing both transformations was rated higher in suspense than each of the three other versions of the game for two of the three suspenseful goals. This validates IRIS as a viable tool in creating suspense in an interactive text adventure video game when both of the restrictions to the game world are applied.

A. *Western Heist Game Domain*

The IRIS system was provided with a western heist domain where the protagonist is a bank robber. Actions allowed in the domain included move, talk, pick up, give, attack, steal, unlock, blow up, and capture. The important items to gather were the key and the dynamite, which allowed the protagonist to make it past the two locked doors on the way to the treasure room. These items were available at several different places in the world. Once in the treasure room, the protagonist could take the loot and try to leave town. There were numerous characters in the world that could aid or hinder the protagonist and some characters present just to provide background flavor for the town.

The protagonist's objective is to get the loot out of town. The IRIS system identified subgoals in the story where suspense should be introduced. The subgoals are 1) getting the dynamite used to blow up the vault door, 2) getting the key used to open the vault safe, and 3) moving the loot from the vault to the train. There were multiple ways to accomplish these subgoals. To get the dynamite, the player could convince the store owner to give him or her a stick or dynamite, or it could be found when exploring the mines. To get the key, the player could steal it from the bank teller or sneak into his room and steal it. When the player tries to get the loot onto the train, another bandit steals it from you. To get it back, the player can attack the bandit or try to negotiate.

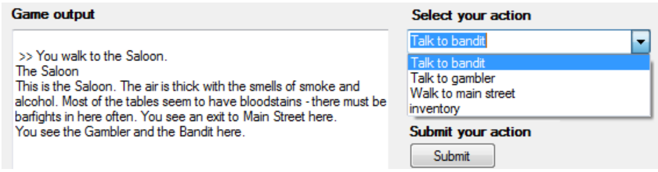

Fig. 14: A screenshot of the GUI that the participants used to play the game.

B. Experimental Design

The story outline that the IRIS system generated with suspenseful conflict around those goals was implemented into a single player Multi User Dungeon (MUD) environment called LambdaMOO. In the game, a participant plays as the bank robber with the objectives described above. A GUI was developed in C# to aid in game play (see Figure 14). The GUI presents a restricted subset of available player actions at a given time, as per restriction 2 previously described. The order in which the actions were presented was randomized after every action selection to prevent priming bias.

Participants were recruited from Computer Science undergraduate classes. There were activities that the students could perform to receive extra credit in the class, and this experiment was one option. The participants came to the researcher's lab for the study where they were first given a consent form to read and sign. Next, they played the game. After the game was over, they were given a survey to measure their sense of suspense around each available game goal. These goals included the three suspenseful goals as identified by IRIS and several others that were not targeted as suspenseful but were made available to increase the number of things to do in the world. Participants rated their agreement on a five point Likert scale with five survey questions that were created from the operationalization of IRIS's definition of suspense. The survey questions are listed in Figure 15. For the survey, the order of the questions regarding the game goals and the questions within each game goal were randomized for each participant. 64 people played the game and all of them completed the post game survey. 17 people each played version 1 (- hints/ - suspense) and version 2 (+ hints / - suspense), and 15 people each played version 3 (- hints / + suspense) and version 4 (+ hints/ + suspense).

A central hypothesis, based on the two proposed restrictions described above, was tested:

Hypothesis: Both 1) an in-game method of giving hints about the ways to achieve the game's goals and 2) the presence of the suspenseful action sequences from IRIS's original story outline and the restriction of player agency during their execution, are necessary to create suspense in an interactive text adventure video game implemented using IRIS's narrative output.

Four different versions of the game, using none, one, or both of the restrictions, were created. Participants were randomly assigned to the different versions. Version 1 contained no game hints or suspenseful sequences, version 2 contained game hints but no suspenseful sequences, version 3 contained no hints but suspenseful sequences and reduced agency during their execution, and version 4 contained both hints and the suspense sequences. It was anticipated that the sense of suspense in the version with both restrictions will be rated higher in the survey than all of the other three versions.

Question 1: I felt excited about the outcome of the goal
Question 2: I felt anxious about the outcome of the goal
Question 3: I felt like an important story event was about to happen
Question 4: I was uncertain about the successful completion of the goal
Question 5: I felt that the outcome would be significant to future events in the story

Fig. 15: The survey questions presented to the participants in the post game survey. These questions were generated by an operationalization of IRIS's definition of suspense.

TABLE III: The results of the Mann-Whitney U tests comparing different versions of the game for each of the three suspenseful goals. The one-tailed Mann-Whitney U test was used in the comparisons involving Version 4 and the two-tailed test for comparisons not involving Version 4. The values in bold indicate significant difference with regards to reported suspense for that survey question, p = 0.05, and the values in parentheses indicate marginal significance, p = 0.10. An asterisk indicates that if a two-tailed test would have been performed, the p value would have been outside the p=0.05 interval.

Version		4v1	4v2	4v3	3v2	3v1	2v1
Key	Q1	**0.006**	**0.01**	**0.003**	0.66	1	0.69
	Q2	0.34	**(0.06)***	**0.03**	0.87	0.12	0.19
	Q3	**(0.09)***	0.17	**(0.07)***	0.58	0.94	0.66
	Q4	**(0.08)***	0.21	0.54	0.35	0.12	0.47
	Q5	**0.05***	1	**0.03**	**(0.06)***	0.88	**(0.08)***
Dynamite	Q1	**0.05***	0.14	0.14	0.75	0.81	0.46
	Q2	0.28	0.85	0.16	**0.009**	0.68	**0.05***
	Q3	**0.03**	0.97	0.96	1	**0.001**	**0.0007**
	Q4	0.23	0.49	0.23	0.48	0.95	0.48
	Q5	0.44	0.96	0.95	1	**(0.10)***	**(0.08)***
Loot	Q1	**0.007**	**(0.09)***	0.15	0.71	**(0.08)***	0.13
	Q2	**(0.10)***	**(0.06)***	**(0.06)***	1	0.67	0.67
	Q3	0.32	0.18	0.84	0.12	0.21	0.64
	Q4	**(0.07)***	0.19	**0.002**	**0.04***	0.19	0.56
	Q5	**0.04***	**0.02**	0.15	0.28	0.48	0.59

C. Results

Using a series of one-tailed Mann-Whitney U tests, version 4 was compared to the other three versions. The other versions were compared among each other using two-tailed Mann-Whitney U tests. This was because no claims about one of these versions being rated higher than the others were made. Mann-Whitney U was used because it tests to see if a measurement from a population of ranked, non-parametric data is significantly different from another. The results of these tests are displayed in Table III.

D. Discussion

Figure 16 shows a visualization of the Mann-Whitney U tests for the "Obtain the key" and "Get the loot onto the train" goals. There are two ways to view the results: comparing responses across versions or across survey questions. When viewing across versions, Version 4 had the most improvement over the versions in this order: Version 1, 3, 2. This is interesting because it suggests that the presence of the

Fig. 16: A visualization of the results of the one-tailed Mann-Whitney U tests comparing Key and Loot goals in Version 4 to those goals in the other three versions. A green box indicates that Version 4 was rated significantly higher with regards to suspense for that survey question, p = 0.05, yellow indicates marginal significance, p = 0.10, and black indicates no significant difference.

suspense sequences (restriction 2) alone is not sufficient to evoke suspense without also providing hints (restriction 1).

When viewing across questions, Version 4 had the most improvement over the other versions in questions 1, 2, and 5. This is encouraging because questions 1 and 2 both directly measure the player's sense of emotion while playing the game, which the full treatment seemed to increase. The lower improvements in questions 3 and 4 may have been caused by player genre expectations. Question 3 was "I felt like an important story event was about to happen." Since players were given these three goals explicitly, they may have felt that, across versions, completing these goals would advance the story. Question 4 was "I was uncertain about the successful completion of the goal." Again, across versions, players may have had the genre expectation that if they are being asked to complete a goal, than there will be some way to complete it.

There was not a significant improvement in Version 4 with regards to suspense for the goal of "Obtain the dynamite." This may be because the actions needed to complete the goal might not have been amenable to plan failure, the method IRIS uses to create suspense. The steps to get the dynamite may have seemed to contain less risk than the steps required to get the key or the loot, decreasing the overall sense of suspense when the player's plan was thwarted. This suggests that in addition to applying the IRIS formula and the restrictions needed to make the experience interactive, the goals that are targeted as suspenseful need to contain high risk actions that will concern the player when plan failure is introduced by the system.

The results of this experiment show that for two of the three goals targeted to be suspenseful, the version of the game with both game creation restrictions in place contained suspenseful story moments and was rated as significantly more suspenseful than the other versions of the game that lacked these restrictions. These restrictions were that players need to be informed of ways to replan towards their goals in the case of plan failure and limitations need to be placed on player actions ensuring that they do not bypass suspenseful story moments. The chosen method of implementing these restrictions was 1) an in-game method of giving hints of how to achieve the game's goals and 2) the presence of the suspenseful action sequences from the original story outline and the restriction of player agency during their execution. This experiment shows that the IRIS system is effective at creating suspense in an interactive text-based adventure game when both restrictions on game world creation are applied.

IX. Conclusion

The IRIS story generation system is a tool to help authors in the creation of suspenseful stories. The author provides initial domain information to the system, which returns a story outline that accomplishes the author's goals and creates suspenseful story moments around those goals. It does this by using a validated definition of suspense and the Artificial Intelligence technique of planning to interject story-level suspenseful conflict around the successful completion of the author's goals. The author can then implement this story outline in a variety of domains. In particular, IRIS was shown to be effective at creating suspense in the domain of non-interactive text stories and a interactive text based video game.

References

[1] Y. Cheong and R. Young, "Narrative Generation for Suspense: Modeling and Evaluation," *Interactive Storytelling*, pp. 144–155, 2008.
[2] V. Propp, *Morphology of the Folktale*. Univ of Texas Pr, 1968.
[3] M. Bratman, "Intention, Plans, and Practical Reason," 1999.
[4] P. Cohen and H. Levesque, "Rational Interaction as the Basis for Communication," 1988.
[5] S. B. Chatman, *Story and Discourse: Narrative Structure in Fiction and Film*. Cornell University Press, 1980.
[6] M. Mateas and P. Sengers, "Narrative Intelligence," 1998.
[7] E. Branigan, *Narrative Comprehension and Film*. Routledge, 1992.
[8] A. Graesser, N. Hoffman, and L. Clark, "Structural Components of Reading Time," *Journal of Verbal Learning and Verbal Behavior*, vol. 19, no. 2, pp. 135–151, 1980.
[9] J. Carbonell, "Subjective Understanding: Computer Models of Belief Systems." Yale Univ Dept of Computer Science, Tech. Rep., 1979.
[10] R. Cullingford, "Sam," in *Readings in Natural Language Processing*. Morgan Kaufmann Publishers Inc., 1986, pp. 627–649.
[11] M. Dyer, *In-depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension*. MIT Press, 1983.
[12] J. Kolodner, *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model*. L. Erlbaum Associates, 1984.
[13] J. Meehan, "Tale-Spin, an Interactive Program That Writes Stories," in *Proceedings of the Fifth International Joint Conference on AI*, 1977.
[14] R. Wilensky, "Pam," *Inside Computer Understanding: Five Programs Plus Miniatures*, pp. 136–179, 1981.
[15] H. Jenkins, "Game Design as Narrative Architecture," *First Person: New Media as Story, Performance, and Game*, 2003.
[16] S. Turner, *The Creative Process: A Computer Model of Storytelling and Creativity*. Lawrence Erlbaum, 1994.
[17] B. R. Tearse, N. Wardrip-Fruin, and M. Mateas, "Minstrel Remixed: Procedurally Generating Stories." in *AIIDE*, 2010.
[18] M. W. Fendt, D. Roberts, and R. M. Young, "Evaluating the Use of Action Templates to Create Suspense in Narrative Planning." in *INT*, 2013.
[19] M. W. Fendt and R. M. Young, "Adapting IRIS, a Non-Interactive Narrative Generation System, to an Interactive Text Adventure game," in *The Twenty-Seventh International Flairs Conference*, 2014.
[20] M. O. Riedl and R. M. Young, "Narrative planning: Balancing Plot and Character," *Journal of AI Research*, vol. 39, no. 1, pp. 217–268, 2010.
[21] J. Bates, "Virtual Reality, Art, and Entertainment," *Presence: The Journal of Teleoperators and Virtual Environments*, vol 1, pp. 133–138.
[22] M. Mateas and A. Stern, "Façade: An Experiment in Building a Fully-realized Interactive Drama," in *GDC*, vol. 2, 2003.
[23] R. Young, "A Developer's Guide to the Longbow Discourse Planning System," *University of Pittsburgh I.S. Program Technical Report*, 1994.
[24] M. O. Riedl and R. M. Young, "Open-world Planning for Story Generation," in *Proceedings of the IJCAI*, 2005, pp. 1719–1720.
[25] T. Trabasso and L. L. Sperry, "Causal Relatedness and Importance of Story Events," *Journal of Memory and Language*, vol. 24, no. 5, pp. 595-611, 1985.
[26] Yannakakis, Georgios N and Martínez, Héctor P, "Ratings are Overrated!." *Frontiers in ICT*, vol. 2, pp 13, 2015.