Generating Stories that Include Failed Actions by Modeling False Character Beliefs

Brandon R. Thorne[1] **and R. Michael Young**[2]

Liquid Narrative Group

[1]Department of Computer Science
North Carolina State University
Raleigh, NC, 27606
brthorne@ncsu.edu

[2]School of Computing
Entertainment Arts and Engineering Program
University of Utah
Salt Lake City, UT, 84112
young@eae.utah.edu

Abstract

Previous work on story planning has lacked a knowledge representation for characters that make mistakes in the execution of their actions. In particular, characters' execution mistakes that arise from errors in belief have not been modeled. In this paper, we describe a state-space planning system and its belief model, together called HEADSPACE, that generates stories that track and manipulates characters' belief about the story world around them. This model is used to produce actions in stories that are attempted but that fail. We show an example story plan that contains failed-action content that cannot be generated by typical planning-based approaches to story creation.

Introduction

In stories, characters commonly attempt to perform actions that fail (Lenhart et al. 2008). For example, when Han Solo is racing to escape the Imperial fleet attacking Hoth in *The Empire Strikes Back (Kurtz et al. 1980)* he pulls the lever to make the jump to hyperspace, but an equipment malfunction in the *Millennium Falcon*'s hyperdrive causes the action to fail both immediately and dangerously.

In examples like this, the failures of character actions aren't simply emergent properties of a complex environment and the limitations of agents operating within it. Rather, the attempts and failures are designed intentionally by authors for narrative effect, e.g., to build tension, to prolong efforts around goal achievement, or to highlight to a reader the disparities of knowledge and ability between characters within the unfolding story world. These roles and others played by action failure within stories are central to many narrative functions. Consequently the development of principled means to generate story lines with failed actions advances the broader goal of automatically creating more natural and compelling narratives.

In recent years, work on automated story generation has shown success developing planning-based generative methods (e.g. (Young et al. 2013; Porteous and Cavazza 2009; Coman and Munoz-Avila 2012)). Planning-based methods for story generation offer a number of attractive features, including guarantees of soundness and completeness and the natural representational fit between plan structures and the goal-directed activity that characters undertake inside narratives. Increasingly, however, researchers have identified limited expressive capabilities in previously developed plan representations when used to characterize story line structure. Knowledge representations that are adequate to produce plans that control robot execution fall short in their characterization of a range of features commonly found in stories. Much work that has gone into plan-based story generation (e.g. (Ware et al. 2014; Bahamón, Barot, and Young 2015; Teutenberg and Porteous 2013)) has sought to retain as many of the benefits of classical planning as possible while also increasing the expressive range of narrative generators.

One limitation of planning approaches arises from their inability to generate plans containing actions that fail. In the work we describe here, we provide the design of an algorithm for story generation that explicitly plans for character actions that fail. The algorithm uses a knowledge representation that provides context for this failure based on the limitations of characters' beliefs about the story world around them (e.g, Han's false belief that the *Falcon's* hyperdrive was operational and could make the jump to lightspeed). We call the pairing of algorithm and its representation defined here HEADSPACE. The HEADSPACE algorithm produces story structure that has many of the advantageous properties found in other plan-based approaches and is more parsimonious than previous approaches to story generation that also address character belief dynamics.

As we describe below, the HEADSPACE system generates stories where

1. agents may operate under mistaken beliefs that lead them to attempt actions which fail, and these attempted actions do not produce the expected effects
2. agents performing actions observe the success or failure of their actions' execution
3. agents revise their belief states in response to
   - an observed failure
   - both passive and active sensing actions

To facilitate the generation of plans that capture these concepts, HEADSPACE draws a distinction between references to conditions that are true in the physical world of the domain and conditions held (or not held) as beliefs of the characters performing the actions.

Related Work

To enrich impoverished plan representations, narrative planning research has incorporated additional constructs into the planning process to support aspects of character decision making. IPOCL (Riedl and Young 2010) adds the concept of intention frames which group a set of actions of one character in furtherance of a single goal. Extending IPOCL, Ware and Young (2011) introduce a model of conflict wherein characters may undertake actions which thwart the intentions of other characters operating in the plan. Bahamón and his collaborators (Bahamón, Barot, and Young 2015; Bahamón and Young 2017) incorporate a model of character personality which influences character behavior at choice points in the narrative. Throughout these extensions to the classical planning algorithm, the previous approaches have assumed two things. First, they make no distinction between the knowledge held by characters and that held by the planning system. Second, the algorithms are based on a long tradition of planning research outside of narrative planning where the soundness of planning algorithms is critical. In typical AI planning, algorithms are shown to be formally sound – that is, that every plan produced by the algorithm is guaranteed to execute correctly. This is a highly desirable property when producing plans for robot execution on a factory floor, but it is limiting when producing plans to drive characters bumbling through a story world.

The initial work addressing disparities of knowledge between agents in a planning context was done by Pollack in her work on the Spirit plan inference system (Pollack 1986). This work in turn motivated Geib's (1994) approach to formalizing intention in a plan generation system. As part of the resulting ItPlanS planner, Geib and Webber (1993) draw the distinction between an action's preconditions and other conditions that are necessary for an action's execution (but are not established by the planner should they not hold). Geib also considers the importance of reasoning about action failure in the context of plan generation. Cavazza and his collaborators (2003) describe an approach to the generation of story sequences where characters are unaware of some aspects of the world around them, including the harmful consequences of some of their own actions.

Teutenberg and Porteous (2013) implement an HSP-style state-space planner (Nebel and Hoffmann 2001) that produces story plans guided by a heuristic that incorporates individual characters' intentions. Instead of using an approach where a central planner generates intentions for all characters, the IMPRACTical planner delegates intentional reasoning to per-agent planning processes. A director agent coordinates suggested actions to best advance a plan toward its goal state.

In an extension to this work, Teutenberg and Porteous (2015) create separate belief models for each of the agent planners. Using a combination of observation axioms and operator annotations, their system can create disparities between the belief models of the agents and the world state. Plan generation is directed based on actions supported by beliefs of the enacting agent. This enables *deceptive social action*, manipulation of the belief state of one agent by actions of another. Subsequently the manipulated agent can be induced to act against its own interests because of the incorrect beliefs it holds.

The approach by Teutenberg and Porteous extends the expressive range of story planners to create stories in which agents undertake actions motivated by intentions that arise from false belief. In their work, they explicitly represent the epistemic state of each character. However, in order for an action to be included in a plan, the preconditions of the action itself must both be believed to be true by the performing agent *and must actually hold in the world at the time the action is attempted*. In situations where false beliefs of an agent suggest that an action be undertaken whose preconditions do not hold in the world, the suggested action is disregarded. Their planning model prevents executable actions from occurring when characters lack the belief in the actions' preconditions. In contrast, we describe here a planning model that enables attempts at actions where the actions are not executable but characters believe that they are.

An alternative approach to the planner-per-agent model is one in which the planning problem is framed as a search through belief space instead of search through state space. In this approach, a state characterization is composed of belief states which are sets of world states described by a given belief model. This approach to plan generation was originally explored through the use of partially observable Markov decision processes; Bonet and Geffner (2000) developed a real-time dynamic programming (RTDP) approach to find preferred paths through the belief space. Bryce and collaborators (2006) provide several metrics upon which to base heuristics for use in such a planner. None of these efforts, however, have targeted story-line generation in the same manner as the work described in the preceding paragraphs (e.g., by adapting other domain-independent planning algorithms to account for the structural properties of stories that are distinct from conventional task representations).

Representation

HEADSPACE uses a PDDL-like (Ghallab et al. 1998) syntax for representing schematized action types in which actions are characterized in terms of *preconditions* – conditions that must obtain in the world state in order for the action to execute – and *effects* – conditions in the world state that change upon the action's successful execution. For efficiency, we follow the approach of Nebel and Hoffmann (2001) and others and pre-compile schematized operators for a given domain into a set of ground operators representing every valid ground instantiation of a domain's act-types. We further differentiate the knowledge representation by describing both preconditions and effects related to the physical world and others that obtain in the beliefs of the character performing the action. In HEADSPACE, a *world frame* captures the sets of ground literals that can be used to characterize the world, as well as the set of symbols used to name the characters capable of taking action in the world.

In the HEADSPACE knowledge representation, the set O contains all the object constants for a given domain. There is a distinguished type of object symbols called *character*, character symbols are contained in a set C where $C \subseteq O$.

Characters are distinguished from other objects by their ability to take action.

In HEADSPACE, a *world frame* captures the sets of ground literals that can be used to characterize the world, as well as the set of symbols used to name the characters capable of taking action in the world.

Definition 1 (World Frame) *A world frame is a tuple $W = \langle GL, C\rangle$ where GL is a set of positive ground literals and C is a set of constants, each denoting a unique character. C contains one distinguished character name E, which designates the environment.*

A *belief state* characterizes the ground literals that a character believes to be true and false, as well as those whose truth values that are unknown to the character.

Definition 2 (Belief State) *Given a world frame $W = \langle GL, C\rangle$, a belief state for some character $c \in C$ is a tuple $BS_c = \langle B_c^+, B_c^-, U_c\rangle$ such that B_c^+, B_c^- and U_c together form a partition of GL, whereB_c^+ designates all the ground literals that c believes to be true, B_c^- includes all the ground literals that c believes to be false and U_c designates all the ground literals that c does not believe to be true and does not believe to be false.*

A *world state* assigns truth values to every ground literal in a world frame, and also provides belief state specifications for every character in a world frame.

Definition 3 (World State) *Given a world frame $W = \langle GL, C\rangle$, a world state is a tuple $w = \langle T_w, F_w, BS_{c_1}, ...BS_{c_n}\rangle$ where T_w and F_w together form a partition of GL, where T_w designates all the ground literals that are true at w, F_w includes all the ground literals that are false at w and each BS_{c_i} designates the belief state for character c_i at w, where $1 \leq i \leq |C|$.*

Definition 4 (Epistemic Goal Specification) *Given a world frame $W = \langle GL, C\rangle$, an epistemic goal specification for some character $c \in C$ is a tuple $\mathcal{EG}_c = \langle B_c^+, B_c^-, U_c\rangle$ such that B_c^+, B_c^- and U_c contain only elements from GL and have no common elements, where B_c^+ designates all the ground literals that c should believe to be true, B_c^- includes all the ground literals that c should believe to be false and U_c designates all the ground literals that c should not believe to be true and should not believe to be false.*

Definition 5 (Master Goal Specification) *Given a world frame $W = \langle GL, C\rangle$, a master goal specification is a tuple $\mathcal{MGS} = \langle T_w, F_w, \mathcal{EG}_{c_1}, ...\mathcal{EG}_{c_n}\rangle$ where each element of the tuple is a set that contains only elements from GL, $T_w \cap F_w = \emptyset$, where T_w designates all the ground literals that must be true at some goal state, F_w includes all the ground literals that must be false at some goal state and each $\mathcal{EG}_{c_i}$ designates the epistemic goal specification that must be true for character c_i at the goal state, where $1 \leq i \leq |C|$.*

A *ground operator* is a complete specification of an action in terms of the character performing the action, the conditions that must be true or false in the world in order for the action to execute, what the character performing the action must believe about the world in order for her to take the action, and how the action, once successfully executed, changes the world and the beliefs of the performing character.

Definition 6 (Ground Operator) *A ground operator GOP is a tuple $GOP = \langle c$,PRE-T,PRE-F,PRE-B^+,PRE-B^-,PRE-U,EFF-T,EFF-F,EFF-B^+,EFF-B^-,EFF-U$\rangle$ such that*

- PRE-T,PRE-F,PRE-B^+,PRE-B^-,PRE-U,EFF-T,EFF-F,EFF-B^+,EFF-B^-,EFF-U $\subseteq GL$
- PRE-T $\cap$ PRE-F=PRE-B^+ $\cap$ PRE-B^- $\cap$ PRE-U=EFF-T$\cap$ EFF-F $\cap$ EFF-B^+ $\cap$ EFF-B^- $\cap$ EFF-U= $\emptyset$
- $c \in C$.

Informally,

- c designates the character initiating (or performing) the ground operator.
- PRE-T indicates the conditions in the world that must be true in order for the operator to execute.
- PRE-F indicates the conditions in the world that must be false in order for the operator to execute.
- PRE-B^+ indicates the conditions that c must believe to be true in the world in order for c to consider the operator executable.
- PRE-B^- indicates the conditions that c must believe to be false in the world in order for c to consider the operator executable
- PRE-U indicates the conditions that c must neither believe to be true or false in the world in order for c to consider the operator executable
- EFF-T indicates the conditions that become true in the world state resulting from the operator's successful execution
- EFF-F indicates the conditions that become false in the world state resulting from the operator's successful execution
- EFF-B^+ indicates the conditions that c believes become true in the world state resulting from the operator's successful execution
- EFF-B^- indicates the conditions that c believes become false in the world state resulting from the operator's successful execution
- EFF-U indicates the conditions that c neither believes are true nor are false in the world state resulting from the operator's successful execution

We call the preconditions and effects that refer to literals that are true or false in the physical world (i.e., PRE-T, PRE-F,EFF-T and EFF-F) as *material* and those that specify beliefs of the character (i.e, PRE-B^+,PRE-B^-,PRE-U,EFF-B^+,EFF-B^-, and EFF-U), as *epistemic*. In HEADSPACE, beliefs are always held by a particular agent, and only about ground literals and their truth values in a particular world. There are no nested beliefs, no existential or universal quantification over beliefs and no implications defined over beliefs.

Following Riedl and Young (2010), the use of the environment as the agent performing an action allows specific types of intentional actions to be inserted into the plan without the need to specify the action's performing character as one of those typically viewed as holding agency in a domain.

We call actions performed by the environment *environmental actions* and actions performed by characters other than the environment *character actions*. To designate environmental actions, we use a naming convention that prepends an asterisk to the front of the action name, For instance, the environmental action that involves striking a character with a lightning bolt might be named *Strike-With-Lightning. For ground operators of environmental actions, the variable c is not used to designate the character performing the action, but rather the character affected by the action. All environmentally performed actions have empty Pre-B^+ and Pre-B^- sets and empty Eff-T and Eff-F sets.

Constructing Story Plans from Planning Problem Specifications

Typical planning representations include a set of schematized action operators characterizing the classes of actions that can occur in a domain. In our approach, we take a set of such operators and a set of object constants and generate a world frame and a set of ground operators from them. This pre-processing is comparable to typical grounding processes used by forward-state planning algorithms (e.g., those of Nebel and Hoffmann (2001)).

A *planning problem*, then, is a tuple including a world frame describing all possible ground literals and characters in a domain, an initial world state characterizing the truth values of all literals and the beliefs of all characters, a goal state giving a partial description of a goal world and a set of ground operators available for characters to execute in the domain.

Definition 7 (Planning Problem) *A planning problem is a tuple $\mathcal{PP} = \langle WF, w_0, \mathcal{MGS}, GO\rangle$ where WF is a world frame, w_0 is a world state defining the initial state of any solution to the problem, $\mathcal{MGS}$ is a master goal specification for the problem, and GO is a set of ground operators for the domain.*

An action, represented by a ground operator, is *executable* in some state w just when all its material preconditions obtain in w.

Definition 8 (Executability/Unexecutability) *A ground operator GOP = $\langle c$,*Pre-T,Pre-F,Pre-B^+,Pre-B^-,Pre-U,Eff-T,Eff-B^+,Eff-B^-,Eff-U$\rangle$ *is executable for character c (or in the case of environmental actions, for the environment) in state $w = \langle T_w, F_w, BS_{c_1}, ...BS_{c_n}\rangle$ just when* Pre-T$\subseteq T_w$ *and* Pre-F$\subseteq F_w$. *We say that a ground operator GOP is unexecutable for character c (or for the environment) in state w just when it is not executable for c in w.*

An action is *apparently executable* in some state w for a character c just when c's belief state in w supports all of the action's epistemic preconditions in w.

Definition 9 (Apparent Executability/Unexecutability) *A character action's ground operator $GOP = \langle c$,*Pre-T,Pre-F,Pre-B^+,Pre-B^-,Pre-U,Eff-T,Eff-B^+,Eff-B^-,Eff-U$\rangle$ *is apparently executable for some character c in state $w = \langle T_w, F_w, BS_{c_1}, ...BS_{c_n}\rangle$, where $BS_c = \langle B_c^+, B_c^-, U_c\rangle$, just when* Pre-$B^+\subseteq B_c^+$,Pre-$B^-\subseteq B_c^-$ *and* Pre-U$\subseteq U_c$.

Definition 10 (Plan) *A plan $\mathcal{P}$ for some planning problem $\mathcal{PP} = \langle WF, w_0, \mathcal{MGS}, GO\rangle$ is a sequence of k tuples $\langle\langle\perp, w_0\rangle, ..., \langle a_k, w_k\rangle\rangle$, where each $a_i \in GO$, $\perp$ indicates a dummy null action placeholder, and each tuple $\langle a_i, w_i\rangle$ indicates the ith action in the plan (attempted in world state w_{i-1}) and w_i the state that obtains after w_i was attempted.*

Definition 11 (Solution) *A solution for some planning problem $\mathcal{PP} = \langle WF, w_0, \mathcal{MGS}, GO\rangle$ is a plan $\mathcal{P}$ for $\mathcal{PP}$ where, for every tuple $\langle a_i, w_i\rangle$ in $\mathcal{P}$, a_i is apparently executable in w_{i-1} and w_i is the world resulting from attempting a_i in w_{i-1}. and for a plan of length k, w_k supports $\mathcal{MGS}$. Specifically, a world state $w_k = \langle T_{w_k}, F_{w_k}, BS_{c_1}, ...BS_{c_n}\rangle$ supports a master goal state $\mathcal{MGS} = \langle T_{MGS}, F_{MGS}, \mathcal{EG}_{c_1}, ...\mathcal{EG}_{c_n}\rangle$ just when $T_{MGS} \subseteq T_{w_k}, F_{MGS} \subseteq F_{w_k}$, and for each $BS_{c_i} = \langle B_{c_i}^+, B_{c_i}^-, U_{c_i}\rangle$ and $\mathcal{EG}_{c_n} = \langle B_{c_i\mathcal{EG}}^+, B_{c_i\mathcal{EG}}^-, U_{c_i\mathcal{EG}}\rangle$, $B_{c_i\mathcal{EG}}^+ \subseteq B_{c_i}^+, B_{c_i\mathcal{EG}}^- \subseteq B_{c_i}^-$, and $U_{c_i\mathcal{EG}} \subseteq U_{c_i}$.*

Plan Generation

The HeadSpace algorithm, shown in Algorithm 1 uses forward-directed state-space search. Search starts at a given initial state, and transition from a given state to its successor states is made through the ground operators that are apparently executable by the characters in the given state. Given a world frame $W = \langle GL, C\rangle$ and a world state $w_i = \langle T_{w_i}, F_{w_i}, BS_{c_1}, ...BS_{c_n}\rangle$, the planner generates successor states for w_i as follows. First, the planner generates the set of all apparently executable ground operators at w_i, designated AE_{w_i}, by taking the union of all actions that appear executable in w_i to each character c_k, $1 \leq k \leq |C|$.

For every executable action in AE_{w_i}, let its successor state w' be equal to modifying w_i as follows:

Definition 12 (Execution Update) *An execution update for action a = $\langle c$,*Pre-T,Pre-F,Pre-B^+,Pre-B^-,Pre-U,Eff-T,Eff-F,Eff-B^+,Eff-B^-,Eff-U$\rangle$ *executed by c in world state $w = \langle T_w, F_w, BS_{c_1}, ...BS_{c_n}\rangle$ creates a new world state $w' = \langle T', F', BS_{c_1}^{'}, ...BS_{c_n}^{'}\rangle$ as follows.*

- $T' = \{T_w \cup$ Eff-T$\}-$Eff-F
- $F' = \{F_w \cup$ Eff-F$\}-$Eff-T
- *For each $BS_{c_i} = \langle B_{c_i}^+, B_{c_i}^-, U_{c_i}\rangle$ in w, create $BS_{c_i}^{'} = \langle B_{c_i}^{+'}, B_{c_i}^{-'}, U_{c_i}^{'}\rangle$ where*
  - $B_{c_i}^{+'} = \{B_{c_i}^+ \cup$ Eff-$B^+\} - \{$Eff-$B^-\cup$ Eff-U$\}$
  - $B_{c_i}^{-'} = B_{c_i}^- \cup$ Eff-$B^-\} - \{$Eff-$B^+\cup$ Eff-U$\}$
  - $U_{c_i}^{'} = \{U_{c_i} \cup$ Eff-U$\} - \{$Eff-$B^+\cup$ Eff-$B^-\}$

For two adjacent tuples $\langle a_i, w_i\rangle$ and $\langle a_{i+1}, w_{i+1}\rangle$ in a plan P, when a_{i+1} is an executable action, we say that a_{i+1} was *executed* by c_{i+1} in w_i, resulting in w_{i+1}. For two adjacent tuples $\langle a_i, w_i\rangle$ and $\langle a_{i+1}, w_{i+1}\rangle$ in a plan P, when a_{i+1} is an unexecutable action, we say that a_{i+1} was *attempted* by c_{i+1} in w_i, resulting in w_{i+1}.

When unexecutable actions are attempted by a character, the actions fail. We call the manner in which the planner manages this kind of action failure the planner's *failure policy*. In the current work, we define a relatively straightforward failure policy. First, with regard to *action occurrence*, none of the attempted action's effects obtain and no material conditions in the world change. In effect, the action does not execute. Second, with respect to *failure detection*, the character executing the failed action immediately detects that it fails, but no other character detects the failure. Third, with respect to *local attribution*, the character executing the failed action assumes that the failure was due neither to execution error nor to an error in the definition of the ground operator. Rather, the character assumes that the failure was due to one or more of the action's epistemic preconditions not holding in the action's world state. Formally, an *epistemic update* occurs when an action is attempted but fails:

Definition 13 (Epistemic Update) *An epistemic update for action* $a = \langle c$,Pre-T,Pre-F,Pre-B^+,Pre-B^-,Pre-U,Eff-T,Eff-F,Eff-B^+,Eff-B^-,Eff-U$\rangle$ *that is attempted in world state* $w = \langle T_w, F_w, BS_{c_1}, ...BS_{c_n}\rangle$ *creates a new world state* $w' = \langle T', F', BS^{'}_{c_1}, ...BS^{'}_{c_n}\rangle$ *as follows.*

- $T' = T_w$
- $F' = F_w$
- *For each* BS_{c_i} = $\langle B^+_{c_i}, B^-_{c_i}, U_{c_i}\rangle$ *in* w*:*
  - *when* $c_i = c$
    - *Create* $BS^{'}_{c_i} = \langle B^{+'}_{c_i}, B^{-'}_{c_i}, U^{'}_{c_i}\rangle$ *where*
      - $B^{+'}_{c_i} = B^+_{c_i}$−Pre-B^+
      - $B^{-'}_{c_i} = B^-_{c_i}$−Pre-B^-
      - $U^{'}_{c_i} = U_{c_i} \cup$ Pre-$B^+\cup$ Pre-$B^-\cup$ Pre-U
  - *otherwise*
    - *Create* $BS^{'}_{c_i} = \langle B^+_{c_i}, B^-_{c_i}, U_{c_i}\rangle$

With respect to *global attribution*, the character doesn't attribute additional causal failure to the world. Finally, with respect to an *epistemic response*, the character comes to not believe all of the epistemic preconditions of the action.

In this initial definition of HeadSpace, we use a simple best-first search approach, though the specification of the algorithm here is meant primarily to demonstrate the functioning of the belief update and executability criteria rather than to advance specific planning heuristics.

Example

Domain Description

To demonstrate the range of belief dynamics and the interaction between belief and execution in HeadSpace, we define a simple Western story domain we call the Break Out domain. Ground operators for the domain are shown in Figure 1, although space limitations required that we list only those ground operators from the domain that are used in the particular plan we examine.

The Break Out domain example makes use of seven operators. They are Draw-Gun, where a character draws a gun from a holster at his or her waist, Fire-At-Lock, where a character fires a gun she's holding at the lock of a door in order to break (and unlock) the door's lock, Check-Cylinder, where a character opens the cylinder of a (revolver) gun that she's holding in her hand, in order to determine if the gun is loaded, Load-Gun, where a character takes bullets from her gun belt and loads a gun that she's already holding in her hand, Open-Door, where a character opens an unlocked door that she's standing near, Traverse, where a character walks through an open door into an adjacent room, Observe-Local+, where a character observes the location of an object that's in the same location as the character, and Observe-Holding+, where a character observes what she is currently holding in her hand.

For the example below, an informal sketch of the planning problem's initial state sets a character, Dolores, locked in a jail cell that has only one exit: a door that's locked. Fortunately, Dolores' compatriot has just tossed a gun belt containing a holstered six-shooter through the cell window, and she has already picked up the belt and strapped it on. The goal for the story is for Dolores to be in the hallway outside her cell.

The plan for the story is shown in Figure 2. In the story plan, Dolores intends to shoot the door's lock to damage it, then open the door and escape her cell. The plan in Figure 2 shows the actual execution of the story fabula. In world state w_0, Dolores believes that her gun is loaded and in her holster, and the door to her jail cell is locked and closed. Her beliefs at w_0 are correct except for the fact that her gun is unloaded. Dolores first draws her gun, then pulls the trigger, intending to shoot the door's lock, thus unlocking it. Because the gun isn't loaded, the action fails. At this point, Dolores

Algorithm 1 HeadSpace algorithm. For Planning problem $PP = \langle WF, w_0, G, GO\rangle$ and plan heuristic ranking function H, call HeadSpace$(WF, H, \langle\langle\bot, w_0\rangle\rangle, G, GO)$.

```
HS($\langle GL, C\rangle$,H,Plans,MGS,GO)
Using heuristic ranking function H, rank all plans in Plans. Let $P$ be the highest ranked plan in Plans.
if $P$ is a solution then
    Return $P$
else
    Let $w$ be $w_k$, the world state in $kth$ (final) tuple in the plan $P$
    Let AE = $\emptyset$
    for all $c \in C$ do
        Let $AE = AE \cup$ all apparently executable actions for $c$ in $w_k$
    end for
    for all $a \in AE$ do
        if $a$ is executable by $c$ in $w_k$ then
            Let $w'$ be the world state resulting from $c$ executing action $a$ in world state $w$
        else
            Let $w'$ be the world state resulting from $c$ attempting action $a$ in world state $w$.
        end if
        Append $\langle a, w'\rangle$ to the end of $P$
        Let $Plans = Plans \cup P$
    end for
    Call HeadSpace$(WF, H, Plans, G, GO)$
end if
```

DRAW-GUN(D,GN)	
PRE-T:	Has(D,GN)
PRE-N:	Holding(D,GN)
PRE-B^+:	Has(D,GN)
PRE-B^-:	Holding(D,GN)
EFF-T:	Holding(D,GN)
EFF-B^+:	Holding(D,GN)

LOAD(D,GN,AM)	
PRE-T:	Holding(D,GN)
	Has(D,AM)
PRE-F:	Loaded(GN)
PRE-B^+:	Holding(D,GN)
	Has(D,AM)
PRE-B^-:	Loaded(GN)
EFF-T:	Loaded(GN)
EFF-B^+:	Loaded(GN)

FIRE-AT-LOCK(D,GN,JC,DR)	
PRE-T:	Loaded(G)
	Holding(D,GN)
	At(D,JC)
	At(DR,JC)
PRE-B^+:	At(D,JC), At(DR,JC)
	Holding(D,GN)
EFF-F:	Locked(DR)
EFF-B^-:	Locked(DR)

OPEN-DOOR(D,DR,JC)	
PRE-T:	At(D,JC)
PRE-F:	Open(DR)
	Locked(DR)
PRE-B^+:	At(D,JC)
PRE-B^-:	Locked(DR)
EFF-F:	Locked(DR)
	Open(DR)

CHECK-CYLINDER-(D,GN)	
PRE-T:	Holding(D,GN)
PRE-F:	Loaded(GN)
PRE-B^+:	Holding(D,GN)
EFF-B^-:	Loaded(D,GN)

TRAVERSE(D,DR,JC,H)	
PRE-T:	At(D,JC)
	Conn(DR,JC,H)
PRE-F:	Closed(DR)
PRE-B^+:	At(D,JC)
	At(DR,JC)
	Open(DR)
EFF-T:	At(D,H)
EFF-F:	At(D,JC)
EFF-B^+:	At(D,H)

*OBSERVE-HOLDING+(D,GN)	
PRE-T:	Holding(D,GN)
EFF-B^+:	Holding(GN)

*OBSERVE-LOCAL+(D,DR,JC)	
PRE-T:	At(D,JC)
	At(DR,JC)
EFF-B^+:	At(DR,JC)

*OBSERVE-LOCAL+(D,D,JC)	
PRE-T:	At(D,JC)
	At(D,JC)
EFF-B^+:	At(D,JC)

Figure 1: Ground Operators used in the Break Out Planning Domain. Here, object constants have been abbreviated to preserve space. Throughout, we use D for Dolores, GN for her gun, JC for the jail cell where she is imprisoned, H for the hallway outside the jail cell, DR for the door between the jail cell and the hallway, AM for the gun's ammo. In this domain, doors are At both of the rooms that they connect.

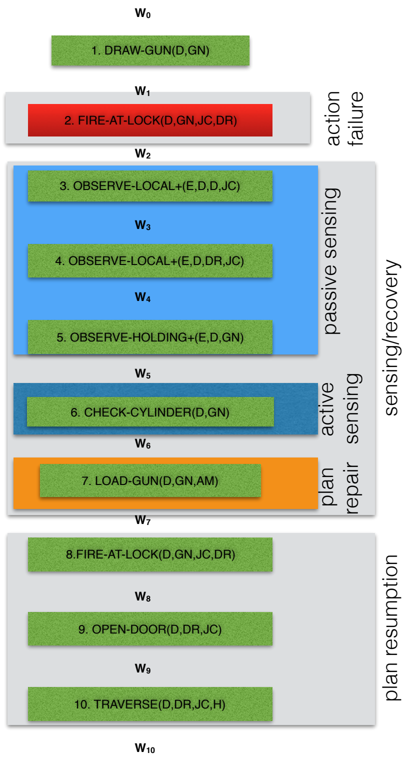

Figure 2: A solution plan for the Break Out domain, showing the sequence of world states and actions attempted and performed by Dolores. Green actions are successfully performed actions. Red actions are ones that are attempted but that fail because their non-belief preconditions are not all met in the world state where they are attempted.

realizes that the action failed, and becomes uncertain about just those beliefs that were involved in the failed action's preconditions.

In the resulting state, w_2, all of the epistemic preconditions for Dolores' execution of Action 2 (the first FIRE-GUN action) have been asserted as unknown in her belief model. Dolores then passively senses the location of the door to her jail cell (Action 3) and her own location (Action 4), and then passively senses that she's holding her gun in her hand (Action 5). She then actively seeks new beliefs about the gun's ammo status by checking the gun's cylinder (Action 6). As a result of Action 6, Dolores believes in w_6 that her gun is unloaded. In Action 7 she takes bullets from her gun belt and loads the gun, then in Action 8 she fires at the lock again. Succeeding this time, the door is now unlocked. Since Dolores believes correctly in w_9 that the door is unlocked, she opens the door (Action 9) and walks out to escape her cell (Action 10).

Discussion and Future Work

The HEADSPACE planning algorithm provides an initial definition of a knowledge representation and planning algorithm to generate plots containing actions that fail due to characters' false beliefs. When specifying the representation and its use, however, we identified a number of larger issues around the generation of plans containing failed actions that remain to be explored.

From a plan generation perspective, this work exposes a significant difference in the construction of heuristic functions for HSP-like domain-independent planners and planners operating in narrative domains. Conventional planners use heuristic functions that are optimized to search for shortest-length plans and do so by searching through a relaxed domain model where some harmful interactions between actions are ignored. Because characters often do not select optimal courses of action in stories, efficient generation of plans that contain these narrative features requires methods for search that are informed by narrative's unique structural needs (similar to the approach taken by Ware and Young (2014)).

Our current method for updating characters' belief states after failed beliefs have at least four limitations that we are currently addressing in near-term work. First, when an action fails, the performing character transfers all preconditions of the failed action into the unknown partition of their belief state. This has the effect, for instance, of Dolores doubting her location and the door's location after the gun fails to go off. A more informed credit assignment algorithm could do better at picking which preconditions should be called into question upon failure of an action.

Second, characters other than the ones performing actions do not become aware of the actions' success or failure. All characters have the same opportunity to sense the world after actions occur, and can thus maintain correct beliefs about the current state of the world, but can't avail themselves of the inferences about epistemic preconditions that arise from the failure of an apparently executable action.

Third, because preconditions and effects do not make reference to actions, characters currently do not become aware of actions as they execute (or are attempted). In future work, we will extend the language around beliefs to include references to action execution.

Finally, beyond the representation and reasoning about belief and action, we are actively extending HEADSPACE to characterize character intentions and the interaction between a character's beliefs, desires and intentions within a story (Young 2017).

Acknowledgements

This material is based upon work supported in whole or in part with funding from the Laboratory for Analytic Sciences (LAS). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the LAS and/or any agency or entity of the United States Government. Further, this material is based upon work supported by the National Science Foundation under Grant No. 1654651. We would like to thank Rogelio E. Cardona-Rivera and Nathan Brown their for insight into action failure in *The Empire Strikes Back*.

References

Bahamón, J. C., and Young, R. M. 2017. An empirical evaluation of a generative method for the expression of personality traits through action choice. In *Proceedings of the Annual Conference on Artificial Intelligence And Interactive Digital Entertainment*.

Bahamón, J. C.; Barot, C.; and Young, R. M. 2015. A goal-based model of personality for planning-based narrative generation. In *Proceedings of the Annual Conference of the Association for the Advancement of Artificial Intelligence*, 4142–4143.

Bonet, B., and Geffner, H. 2000. Planning with incomplete information as heuristic search in belief space. In *Proceedings of the Annual Conference of the Association for the Advancement of Artificial Intelligence*, 52–61. AAAI Press.

Bryce, D.; Kambhampati, S.; and Smith, D. E. 2006. Planning graph heuristics for belief space search. *Journal of Artificial Intelligence Research (JAIR)* 26(1):35–99.

Cavazza, M.; Charles, F.; and Mead, S. 2003. Intelligent virtual actors that plan... to fail. In *Smart Graphics*, 343–368. Springer.

Coman, A., and Munoz-Avila, H. 2012. Creating diverse storylines by reusing plan-based stories. In *Working Notes of the ICCBR-12 Workshop on TRUE and Story Cases: Traces for Reusing Users' Experiences - Cases, Episodes, and Stories*.

Geib, C., and Webber, B. 1993. A consequence of incorporating intentions in means-end planning. In *Working Notes–AAAI Spring Symposium Series: Foundations of Automatic Planning: The Classical Approach and Beyond*.

Geib, C. W. 1994. The intentional planning system: ItPlanS. In *Proceedings of the International Conference on AI Planning Systems*, 55–60.

Ghallab, M.; Howe, A.; Christianson, D.; McDermott, D.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. Pddl—the planning domain definition language. *Proceedings of the AI and Planning Systems Conference* 78(4):1–27.

Kurtz, G.; Brackett, L.; Kasdan, L.; Kershner, I.; Lucas, G.; Hamill, M.; Ford, H.; Fisher, C.; Williams, B. D.; Daniels, A. D.; and Williams, J. 1980. *The Empire Strikes Back*. United States: 20th Century Fox.

Lenhart, A.; Kahne, J.; Middaugh, E.; MacGill, A. amd Evans, C.; and Vitak, J. 2008. *Teens, video games, and civics*. Pew Internet and American Life Project.

Nebel, B., and Hoffmann, J. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 14:253–302.

Pollack, M. 1986. A model of plan inference that distinguishes between actors and observers. In *Proceedings of the 24th annual meeting of the ACL*.

Porteous, J., and Cavazza, M. 2009. Controlling narrative generation with planning trajectories: the role of constraint. In *Proceedings of Second International Conference on Interactive Digital Storytelling (ICIDS)*.

Riedl, M., and Young, R. M. 2010. Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research* 39(1):217–268.

Teutenberg, J., and Porteous, J. 2013. Efficient intent-based narrative generation using multiple planning agents. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 603–610. International Foundation for Autonomous Agents and Multiagent Systems.

Teutenberg, J., and Porteous, J. 2015. Incorporating global and local knowledge in intentional narrative planning. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 1539–1546. International Foundation for Autonomous Agents and Multiagent Systems.

Thomas, J., and Young, R. M. 2010. Annie: Automated generation of adaptive learner guidance for fun serious games. *IEEE Transactions on Learning Technologies* 3(4):329–343.

Ware, S., and Young, R. M. 2011. CPOCL: A narrative planner supporting conflict. In *Proceedings, Seventh International Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Ware, S., and Young, R. 2014. Glaive: a state-space narrative planner supporting intentionality and conflict. In *Proceedings of the International Conference on AI and Interactive Digital Entertainment*, 80–86.

Ware, S. G.; Young, R. M.; Harrison, B.; and Roberts, D. L. 2014. A computational model of narrative conflict at the fabula level. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games* 6(3):271–288.

Young, R. M.; Ware, S. G.; Cassell, B. A.; and Robertson, J. 2013. Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives. *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative* 37(1-2):41–64.

Young, R. M. 2017. Sketching a generative model of intention management for characters in stories: Adding intention management to a belief-driven story planning algorithms. In *Working Notes of the AIIDE Workshop on Intelligent Narrative Technologies*.