Generating QUEST Representations for Narrative Plans Consisting of Failed Actions

RUSHIT SANGHRAJKA[*], School of Computing, University of Utah, USA
ERIC W. LANG, School of Computing, University of Utah, USA
R. MICHAEL YOUNG, Entertainment Arts and Engineering Program and School of Computing, University of Utah, USA

Increasingly, research on narrative planning is expanding the expressive range of narrative generation systems, producing plot lines with structures like failed action, mistaken character belief and the integration of authorial and character-centered plans and intentions. Evaluation of these systems' expressive capabilities is essential to determining their strengths. Some prior evaluative methods have measured the efficacy of narrative planners by characterizing a user's experience during generated narratives. These approaches have focused on comparing the mental model a user forms during the experience of a narrative with the plan data structure that served as the basis of the narrative's plot, but have not considered evaluating the user's understanding of failed actions in narrative. To that end, we propose an algorithm to translate plans containing failed actions into a commonly used cognitive mode of narrative comprehension called QUEST. We then sketch how this translation will play a role in a planned evaluation of users' experiences reading stories produced by narrative planning systems that generate stories with failed actions.

CCS Concepts: • **Computing methodologies → Artificial intelligence**; **Reasoning about belief and knowledge**.

Additional Key Words and Phrases: narrative planning, narrative comprehension, cognitive models, expressive range, failed actions

ACM Reference Format:
Rushit Sanghrajka, Eric W. Lang, and R. Michael Young. 2021. Generating QUEST Representations for Narrative Plans Consisting of Failed Actions. In . ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Foundational research in cognitive psychology [5, 15, 31] has shown that effective narrative comprehension is essential to a user's experience of a story. While this is true for human-produced narratives like films and novels, it is also true for automatically generated narratives produced algorithms exhibiting a range of different degrees of artificial intelligence, like interactive video [24], video games which allow the user to participate in their storytelling for an immersive experience, and interactive narratives [7, 18]. In such systems, a user's ability to understand the functional elements of a story is one of the central contributors to their experience, and systems that can have access to a model of users' understanding can leverage that model to target specific types of user experience.

Narrative planning algorithms have made significant advances in the computational modelling of narratives. Plan-based approaches such as IPOCL [23], Glaive [32] and IMPRACTical [27] have shown promise in producing

[*]All authors contributed equally to this research.

FDG'21, August 2–6, 2021, Online

https://doi.org/10.1145/1122445.1122456

stories with understandable structures such as intentionality and conflict. Narrative modelling approaches have been effective in generating narrative which can then be used in interactive systems as well [1, 7]. These approaches attempt to computationally characterize the cognitive impact of stories.

While these planning approaches can be used to generate unique narratives, the story itself should be effective in being reliably comprehensible to users as intended by the author. Evaluation strategies for these planners often use cognitive models of comprehension that are based on how humans process narrative events to evaluate whether people understand the causality of events as generated by these systems [6, 23]. As narrative planning algorithms increase their expressive range to produce a wider set of story elements, methods used to evaluate story generators must be developed that can gauge the efficacy of generators creating new forms of stories.

One such cognitive model, called QUEST [16, 17], has been developed and validated as a cognitive model of human question answering in the context of stories. The model uses a graph-form knowledge representation – called a QUEST Knowledge Structure, or QKS – to characterize the relationships between events, goals and intentions that underlie a story. In the QUEST model, the arc distance between any two nodes in a QKS predicts how well the second node serves as an answer for *why*, *how*, or *in-order-to* questions about the first node. The model has been used by the narrative planning community to evaluate their planning approaches [3, 11, 22, 32]. In these methods, plan data structures are translated into the data structures representing a cognitive model of narrative comprehension. Experiments compare the way humans process the generated stories to the predictions made by the cognitive models. However, translation algorithms used to generate QUEST graph structures from narrative planning output have only focused on positive actions in the plan [6, 11] (e.g., actions that are attempted by characters and execute correctly in the story world).

New algorithms that create narratives with greater expressive range have prompted consideration of adapting or extending evaluation methods, including those centered around user experience. In particular, a need has arisen because of increased expressivity around failed actions and mistaken beliefs [20, 30], two narrative elements not accounted for in prior evaluative methods. Evaluation methods that adopt a cognitive stance require new knowledge representations that include failed actions in their model.

While not used by previous approaches of evaluation, the original QUEST Knowledge Structure (QKS) theories consider the possibility of failed events and unachieved goals appearing in the story discourse. To date, however, no evaluative approaches that leverage QKSs have taken advantage of these elements to gauge the effectiveness of stories with failed actions. This paper briefly describes the character of QKSs containing failed actions and their semantics as defined by the original QUEST work. We then present an automatic translation algorithm to translate from a modified Planning Domain Definition Language (PDDL) representation [14] into QKS that can be applied in evaluation strategies. Finally, we describe future work on evaluation methods that can make use of this translation process. These evaluation methods help understand the user experience of the narrative.

2 RELATED WORK

2.1 Narrative Planning

There has been a significant amount of work on narrative generation that make use of AI planning algorithms with extended knowledge representations intended to increase the expressive range [26] of these generative systems relative to conventional planners. The approaches accomplish this by considering aspects such as tension [12], suspense [9, 10, 19], character personality and affect [4, 13], intentionality [23, 27] and nested belief [25] representations in the narrative planning process. Recent work in narrative generation has also looked at producing narratives with failed actions. Porteous and Lindsay [20] have looked at non-cooperative narrative planning where characters act to sabotage each others' plans. In their model, plan sabotage results in characters having to reassess the world and come up with ways to achieve goals in a competitive environment. Teutenberg and Porteous [28] also describe characters' belief models and characters acting on incorrect beliefs in their work in the context of deceitful actions. While these two approaches do not specifically address characters acting and *failing*, they expand upon the expressivity of narrative generation to include representation and behavior based on incorrect beliefs as well as plan updates prompted by characters' discovery of plan failure.

Shirvani and their collaborators [25] take a different approach to failed actions. They adopt a belief model that leverages possible worlds to represent nested epistemic beliefs in their knowledge representation. They extend typical possible worlds models by including, among other elements, surprise elements that represent situations where an action has unexpected effects in the world relative to a character's belief. Their primary focus is on actions that, when executed, have unexpected outcomes. However, their notion of surprise also encompasses actions that characters believe will execute correctly but instead fail due to the characters' incorrect beliefs. Their approach results in a greater expressive range for narratives by including, among other characteristics, a representation for failed actions.

Thorne and Young [30] define a knowledge representation that explicitly includes narrative plans in which characters can attempt actions that fail due to mistaken beliefs held by the character performing the action. This work expands the space of stories that can be generated using computational models of narrative planning by incorporating character beliefs and requirements for attempted execution for each action based on those beliefs. During plan construction, the planner can add to a plan steps whose preconditions are not met (when, for instance, a character's mistaken beliefs support their attempt to execute the step). These steps will fail, and so in anticipation, the planner marks them as such, substituting into the plan alternative, failed versions of those actions. Subsequent actions for characters that observe action failures may then address the observer's misconceptions and repair those characters' plans.

2.2 QUEST: A Model of Question-Answering

QUEST [16, 17] is a model of human question-answering in the context of stories. The QUEST model relies on graphical representations of knowledge known as QUEST knowledge structures (QKSs). These QKSs are built from story structure and have been experimentally evaluated to correlate with aspects of human narrative comprehension. QUEST knowledge structures use nodes for events, goals and states, and edges used to represent causal, intentional, temporal or other types of relationships between these nodes. Questions above event nodes (e.g., why did an event happen, how did an event happen, what were the consequences of an event happening) can potentially be answered by other nodes in the graph (e.g., From 'What is the consequence of event A happening?" to "State S is a consequence of event A.") Question-answer pairs can be generated from the knowledge graph by selecting two nodes of the appropriate types: one node to serve as the source of the question and another node to serve as the source of its potential answer. The QUEST model defines how the arc-distance between the question and answer node is calculated from the graph, starting from the question node as the source and finding possible target nodes as answers. This arc distance acts as a predictor of how strongly a human reader will rate the answer node as an effective answer to the question. The arc search procedures between question nodes and answer nodes vary depending on question type, and are primarily sensitive to arc direction and arc label.

As mentioned above, the QUEST knowledge structure consists of a directed graph with labeled nodes and arcs. There are four types of nodes: event, state, goal and style. Nodes represent proposition-like expressions which can contain a predicate (i.e. verb or adjective) and one or more arguments (nouns, embedded propositions). A state node represents an ongoing characteristic which remains unchanged within the timeframe it is presupposed. An event is a state change that occurs within the timeframe. A goal refers to a state or an event that the agent desires. This work looks at QUEST graph structures primarily in the context of goal hierarchies as defined by [16, 17] and hence style nodes are not considered. Table 1 provides a concise definition of some of the arcs that are present within QKSs and are important in the context of this work.

Arc Type	Definition	Composition Rule
Consequence (C)	A causes or enables B A precedes B in time	Event | State -C⟶ Event | State
Reason (R)	B is a reason or motive for A B is a superordinate goal for A A is achieved before B is achieved	Goal -R⟶ Goal
Outcome (O)	B specifies whether or not the goal A is achieved	Goal -O⟶ Event | State
Initiate (I)	A initiates or triggers the goal in B A precedes B in time	Event | State -I⟶ Goal

Table 1. Selected set of arcs within the QUEST Knowledge structures as defined by Graesser et al. [16]

2.2.1 Arc Search Procedures in QUEST Knowledge Structures. For goal-oriented substructures, Graesser [17] defines the arc search procedures for "Why", "How", and "What are the consequences of" questions. We describe the arc search procedures for these questions and then use the arc search procedures on the proposed algorithm to show that the QKS created is effective at finding possible answers to questions, and the arc search procedure rates them in a way consistent with human comprehension.

The "Why" questions have four sets of nodes produced as answers when probed on a goal node G. They are (1) superordinate goals via paths of forward Reason and backward Manner arcs, (2) sibling Goal nodes via paths of forward before-arcs, (3) goal initiators connected to G or G's superordinate goals by a backward Initiate arc, and (4) causal antecedents to each goal initiator which initiate from the goal initiator via backward Consequence arcs, Implies arcs, backward Outcome arcs, and backward Initiate arcs.

The "How" questions have likely answers on nodes that are found by traversing through backward Reason arcs and forward Manner arcs.

"What are the consequences" questions can be answered in two ways. The first way involves traversing through forward Reason arcs and backward Manner arcs to find achieved superordinate goals. The second involves finding causal consequences of the queried node and the superordinate goals by following forward Consequence, Implies, Outcome, and Initiate arcs.

2.2.2 Evaluating User Experience Leveraging QUEST to Gauge Narrative Comprehension. Christian and Young [11] proposed the first approach for translating a narrative plan generated by an AI planner into a QUEST knowledge graph. Their process was automatic and was evaluated to show that the approach was successful at predicting user comprehension in a story. Additionally, Riedl used QUEST as the cognitive model for evaluating the IPOCL planner in their dissertation work [22].

Cardona-Rivera and their collaborators [6] studied the QUEST model with respect to intention-based partial order causal link (POCL) plans, summarizing the approaches by Christian and Young [11] as well as Riedl and Young [23]. They proposed another algorithm to generate QUEST knowledge structures which builds off of previous approaches. However, existing approaches that translate the output of narrative planning systems into QKS structures are limited. Their source plans do not contain failed actions, and so their translation processes do not address elements that are increasingly present in more expressive narrative plan generators (e.g., those planners described in Section 2.1 above). As we describe below, QUEST already has a methodology for representation of failed actions, and this work aims to provide a way to translate narrative planners into QUEST knowledge graphs which can represent failed actions.

3 FAILED ACTIONS, QUEST KNOWLEDGE STRUCTURES, AND PDDL PLANS

3.1 Failed Actions in QUEST Knowledge Structures

In this section, we describe those portions of the QUEST knowledge structure that are related to failed actions and unachieved goals – both elements that have not been previously used in QUEST-based evaluations of user experience in narrative generation systems. Full definitions for the QUEST knowledge structures are provided in [16] along with examples and definitions for the components of the graph structure that models knowledge built from a human's reading of a narrative text. In brief, an event is a primitive that represents the change in state; an action is composed of an event paired with a goal node using an outcome arc.

Failed Events. Event nodes are one of the three types of nodes in the conceptual graph structures defined by Graesser and Clark. These nodes are used to represent two types of actions. First and most commonly, they represent events that take place in the world and cause the world to change in state. Second, the representation can also be used to represent failed events in the QUEST graph structure. Failed events represent an event where there was an unsuccessful attempt to change the world.

Unachieved Goals. Unachieved goal nodes are used in QUEST to represent goals that are not achieved in a story. We divide these goals into two categories. In the first, goals could be unachieved because the characters never attempted to execute actions that would have led to their achievement in the story segment. These types of unachieved goals are represented as Goal nodes that have no Outcome edges coming out of them. In our second category, goals can be unachieved because, through the character's actions, they realize that it is impossible to achieve them (either by observation or through a failed action). One such example of this case is shown by Graesser and Clark in a knowledge structure for *REWARDING*. This is shown in Figure 1. In this figure, Node 2 is an example of a goal node that was unachieved because the event sequence does not show any actions that would have led to its achievement. Node 2 is a goal node for "X wants to get a reward", and the graph structure does not specify whether it was achieved due to the outcome not being determinate. On the other hand, Node 10 is an example of an unachieved goal node by attempting it. Node 10 is a subordinate goal node that represents X's goal of "informing Z to stop threatening Y". The goal node has an outcome edge directed to a state node which depicts that, due to the outcome node 11 ("It is impossible to inform Z to stop threatening Y"), X's goal of informing Z to stop threatening Y failed. In this particular example, upon failing to achieve their sub-goal, X resorts to another plan to achieve their super-ordinate goal (Goal 9: X wants to stop Z from threatening Y), which can be observed in goal node 12 "X wants to kill Z".

Failed Actions. There is no explicit definition for failed actions provided by Graesser and Clark. Actions performed by an agent could fail due to many reasons. An action could fail simply because of how it is performed (such as making a three point shot in basketball), fail because it has a probabilistic chance of success (such as rolling a six on a dice), or fail because the actor had incorrect beliefs about the world (such as attempting to open a door when it is locked). **Intentional actions** are defined as goal nodes linked to event nodes with a positive outcome arc. Based on this, we determined that failed actions must be goal nodes connected to an event node with a negative outcome arc, i.e. a character performed an event with a goal which did not go as they intended it to.

Negative Outcomes. Negative Outcome arcs are defined explicitly by Graesser and Clark as occurring when the outcome node directly clashes with the goal node." Outcome arcs can link to both event and state nodes, so negative outcomes can lead to either events or states. Graesser and Clark provide an example of a negative outcome to a state node shown in the arc between Nodes 10 and 11 in Figure 1.

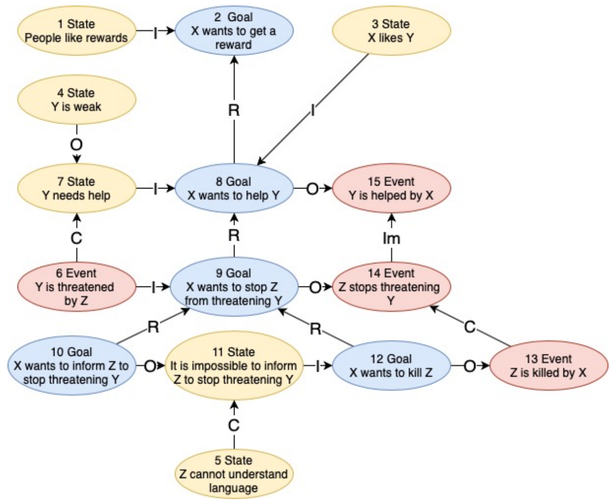

Fig. 1. The conceptual graph structure for REWARDING, as depicted by Graesser and Clark. Nodes 2 and 10 are examples of unachieved goal nodes. Arcs labeled with "R" are reason arcs, "O" are outcome arcs, "C" are consequence arcs, and "I" is the initiate arc. Yellow nodes denote states, blue nodes denote goals, and red nodes denote events.

3.2 Towards the Automatic Creation of QUEST Knowledge Structures from Plans That Include Failed Actions

In order to incorporate failed actions in QUEST knowledge structures in a way that can answer goal-related questions, we must ensure that the structure can appropriately answer all three questions above: "how", "why", and "what are the consequences".

Christian and Young [11] defined the initial approach to generate QUEST knowledge structures automatically from plans generated from narrative planners and used that method to empirically evaluate viewers' comprehension of a cinematic narrative. One limitation of Christian and Young's algorithm is that it did not consider failed actions and therefore would not produce a sensible QKS for a plan incorporating failed actions. Essentially, the reason arcs from goals corresponding to failed actions within a QKS generated by their algorithm would cause incorrect question answering for the failed action's goal. As an example, if a story plan involved a character attempting to do action A and failing, with the character then proceeding onto performing action B, the goal structure would read, "the character wanted to do A in order to do B". If a character tried to open a door without knowing it was locked, failed in its attempt, realized the door was locked, and then unlocked it, for example, this would read as "the character wanted to open the door in order to unlock the door." Instead, the question of "why did the character try to open the door?" is better asked through character intent, not by what happens next in the finished plan.

Our proposed algorithm, shown in Algorithm 1, addresses the above issues and allows for the generation of new types of QKS structures that are both faithful to Graesser and Clark's definitions and also incorporate failed actions effectively.[1] When generating a QKS with failed events, the proposed algorithm requires a knowledge representation that includes both the story plan (actions that are executed or attempted in the story world), as well as a model of the plans that are *intended* by each character and the beliefs held by those characters that

[1]Our worked example shows a plan where characters' actions fail due to incorrect beliefs. In this plan, the characters' beliefs are automatically updated when they observe their own action failures. This is consistent with plans produced by the HeadSpace planner [29], for instance, but is not a required aspect of the translation algorithm we present here.

support the executability of their plans. The plans held in the mental state of each character, called intention plans [6], are held over a set of contiguous states, and are consistent with the character's beliefs during the period when they're held. Intention plans are held by characters until either the goals of the intention plans are achieved by steps in the actual story plan or until a failed action in the story plan prompts a character to revise their beliefs in a way that makes the intention plan inconsistent. The definition of an intention plan is provided below, based on Cardona-Rivera et al's [6] plan definition.

Definition 1 (Intention Plan). *An Intention Plan is a tuple $(S, B, \prec, L)$ where S is a set of steps, B is a set of variable bindings, $\prec$ is a set of orderings, and L is a set of causal links. The intention plan must, on the basis of the belief state of the character rather than the actual world state, fulfill typical plan completeness criteria:*

- *For every precondition p of every step $u \in S$, there exists a casual link $s \xrightarrow{p} u \in L$ (i.e., every precondition of every step is satisfied).*
- *For every causal link $s \xrightarrow{p} u \in L$, there is no step $t \in S$ which has effect $\neg p$ such that $s \prec t \prec u$ is a valid ordering according to the constraints in $\prec$. In other words, it is not possible that a causal link can be made undone before it is needed.*

Our translation algorithm incorporates the intention plan of the character as a chain of goals with reason arcs so that forward goal relationships are maintained from failed actions. The intention plan incorporated in this algorithm does not stop when a goal is not achieved because of a failed action. This is consistent with Graesser's system, as questions such as "what are the consequences" solely look at achieved goals, not unachieved goals, while questions like "why" can traverse reason arcs of both achieved and unachieved goals.

Examples of the way candidate answers for specific questions can be computed using the output from our algorithm can be found in Section 4.2 below.

4 EXAMPLE TRANSLATION: ESCAPE SCENARIO

The escape scenario is a simple domain and problem constructed to help describe how the QKS generation algorithm works. In the escape scenario, there is an agent in a room with two doors: the west door and the east door. The agent is on the west side of the room. The agent believes that both doors are unlocked, but in reality only the east door is unlocked. The agent is trying to escape from the room and either door is suitable for the escape. The agent only has three actions: moving from one side of the room to the other, opening unlocked doors, and escaping through an open exit door. The full story plan and all intention plans are shown in Figure 2.

When the planner constructs the agent's first intention plan, it creates the simple plan of opening the west door and then escaping through the door. This is consistent with the agent's (mistaken) beliefs indicating the west door is unlocked. Steps from this plan are added to the story plan up to the point where the first action fails (in this case, at the agent's first action, Step 8). At this point, the agent's mistaken beliefs are updated and its inconsistent intention plan is discarded in favor of a new one consistent with its updated beliefs. Because that intention plan has no failed actions, all its steps are added to the story plan. The final plan has four steps: the agent attempts to open the west door and fails, the agent moves to the east location, the agent opens the east door, and then the agent escapes.

4.1 Constructing a QKS

To create a QKS corresponding to the plan in Figure 2, the translation algorithm begins by creating an event node for every step in the story plan. This results in four events corresponding to the four steps described above (line 3). The next part of the algorithm (lines 4–7) generates a state node for every effect of each event along with a consequence link from the event to the generated state node. For example, the event of the agent moving to the east side is linked with a consequence arc to a new state node for `At(agent, east-loc)`. Similarly, the event

Data: A plan P $(s, B, \prec, L, I)$
Result: A QUEST Knowledge Graph

Create a total ordering for all the steps in P;
for *every step* s_i *in P* **do**
    Create event node ε_i for s_i;
    for *each effect* e_j *in* s_i **do**
        Create a State node σ_{e_j} ;
        Link it to ε_i with a Consequence arc $\varepsilon_i \xrightarrow{c} \sigma_j$;
    end
end
$k \leftarrow -1$;
for *each Intention Plan* φ_i *in I* **do**
    $active \leftarrow true$;
    for *each step* s_j *in* φ_i **do**
        Create a goal node Υ_{ij};
        if *active* **then**
            Link Υ_{ij} to event node ε_k using an Outcome arc $\Upsilon_{ij} \xrightarrow{O} \varepsilon_k$;
            if ε_k *was a failed event* **then**
                $active \leftarrow false$;
            end
            $k \leftarrow k + 1$;
        end
    end
    for *each causal link* $s_l \xrightarrow{p} s_m$ *in* φ_i **do**
        Link the corresponding goal nodes with a Reason arc $\Upsilon_{il} \xrightarrow{R} \Upsilon_{im}$;
    end
end
for *each failed event* ε_i **do**
    if *there exists a causal link* $s_i \xrightarrow{p} s_j$ **then**
        Connect the state node for p σ_p to the goal node connected to ε_j with an Initiate arc $\sigma_p \xrightarrow{I} \Upsilon_{xj}$;
    end
    else
        Connect ε_i to the event node ε_j for the agent's next step based on the total ordering with an Initiate arc $\varepsilon_i \xrightarrow{I} \varepsilon_j$;
    end
end
For each causal link $s_i \xrightarrow{p} s_j$ in P, link the corresponding state node σ_p to the event node ε_j with a Consequence arc $\sigma_p \xrightarrow{c} \varepsilon_j$;

Algorithm 1: Translation algorithm for creating a QKS from a Plan

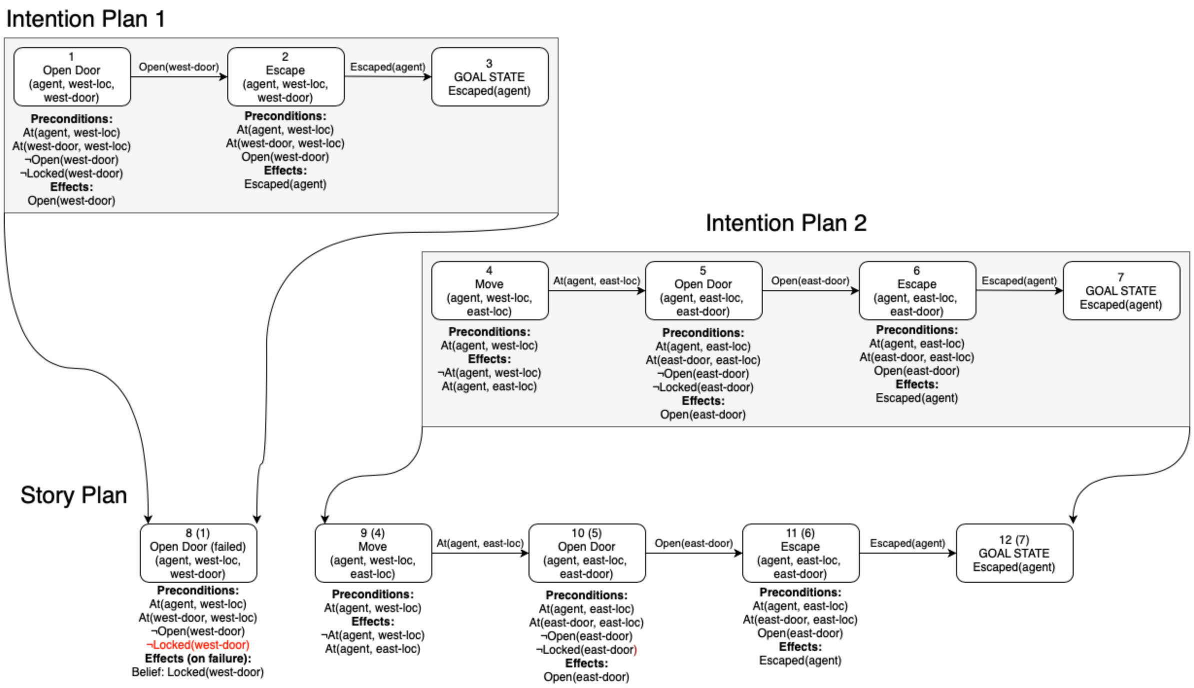

Fig. 2. The character's intention plans and the final story plan for the "Escape" scenario. Steps are indicated using rounded rectangles and arcs between steps indicate causal links, with labels on those links indicating the condition connecting the source and sink steps. Preconditions for each action are showing immediately below the action, and preconditions that match a character's mistaken beliefs are shown in red. Intention plans – those plans formed by characters indicating their intended courses of action, are shown above with gray backgrounds. The story plan is shown below. Arrows from the boundaries of the intention plans into the story plan indicate the span of the story during which the Agent character holds the intention plans. Numbers in each step are for reference, and story plan steps contain both reference numbers and parenthetical numbers linking each step to its origin in an intention plan. Once Step 8 in the story plan fails, the character's beliefs about the West Door being locked are revised, the character's intentions are replanned, and Intention Plan 2 is adopted and added to the story plan.

for failing to open the west door causes a belief update of `Locked(west-door)` for the agent, so a state node is created for that belief update and it is linked from the event node via a consequence arc.

Once all events are added and their effects are represented by state nodes connected from consequence arcs, the algorithm iterates through every intention plan in order that was used to generate the story plan (line 10) while considering each event node that it had added on line 3. The first intention plan was to open the west door and then escape. The algorithm creates a goal for opening the west door (line 13). It then links this goal with an outcome to the event of failing to open the west door (line 15). This intention plan is then flagged as having a failing outcome (line 17), so the remaining steps in the intention plan (in this case, just the escape through the west door) are only added as goals. Once all goals for the intention plan are added, causal links between the steps in the intention plan are translated as reason arcs between goals (line 23). In this case, the "Open West Door" goal links with a reason arc to the "Escape West" goal.

The algorithm proceeds to the next intention plan: to move to the east door, open the east door, and then escape through the east door. For each of these steps it creates a goal and then links the goal with an outcome link to the corresponding event in the final plan. Afterward, it translates any causal links in the intention plan into reason arcs for the goal nodes.

The next step of the translation is to create initiate arcs from failed actions to the first goal of the next intention plan (lines 26–33). These initiate arcs make it clear that the next goal was a product of what happened previously (the observation by a character of a failed action), not a necessary step based on the previous goal, which is why the new intention plan's goal does not have a reason arc from the previous goal corresponding to the failed action. Instead, if the failed step has a causal link to the next step, an initiate arc is created from the failed step's condition that the causal link depends on to the goal of the next step. If no such causal link exists, the goal of the failed event points to the goal of the next step with an initiate arc. In this example, failing to open the west door initiates the goal of moving to the east, as there is no causal link from the failing step to the next step in the plan.

Finally, all causal links in the final plan are represented by adding consequence arcs from the effect of the event that was derived from the source step of the causal link to the event that was derived from the sink step of the causal link (line 34). In our example, one instance of this is the consequence arc from the state `At(agent, east-loc)` to the event corresponding to opening the east door.

The final diagram of our QKS for this example can be seen in Figure 3.

4.2 Escape Scenario Goal Questions

As described in Section 3.2, The QUEST model is capable of taking a QKS and a node from that QKS and returning sets of logically appropriate nodes that can serve as answers to "Why," "How," or "What are the consequences of" questions about the input node. It can also provide relative rankings of goodness of answer within each set. We provide short descriptions below for the processes used in QUEST to determine these sets, including for QKSs that contain failed actions.

Why: "Why" questions about goals start with a goal and traverse reason arcs forward, manner arcs backward, sibling nodes from forward before-arcs, goal initiator arcs backwards, and consequence arcs from the start of those initiator arcs backwards. Among these arcs, we solely include reason arcs, initiator arcs, and consequence arcs; manner arcs do not exist in our system. Our QKS can provide goodness of answer ratings for "why" questions such as:

- *Why does the character want to open the west door?* The character wants to open the west door in order to escape to the west.
- *Why does the character move to the east door?* The character moves to the east door because they failed to open the west door. The character failed to open the west door because they believed the west door was unlocked.

How: "How" questions traverse backward through reason arcs and forward through manner arcs using **achieved** goals. Since our translation algorithm doesn't generate manner arcs, we solely rely on reason arcs. The main benefit of our translation algorithm is in avoiding invalid answers to "How" questions. Namely, by ensuring there is no reason arc from the "Open West Door" goal to the "Move to East Door" goal, we avoid including an answer like "The character wants to open the west door in order to move to the east door" for a question like *How does the character want to move to the east door?*. Such a "how" question response would be included if, for example, we tried to use Christian and Young's [11] algorithm on a plan with failed actions.

What are the consequences: The arc search procedure for these types of questions can only progress through *achieved* goals and are answered by traversing through forward reason arcs, backward manner arcs, forward consequence arcs, forward implies arcs, forward outcome arcs, and forward initiate arcs. In our system, the

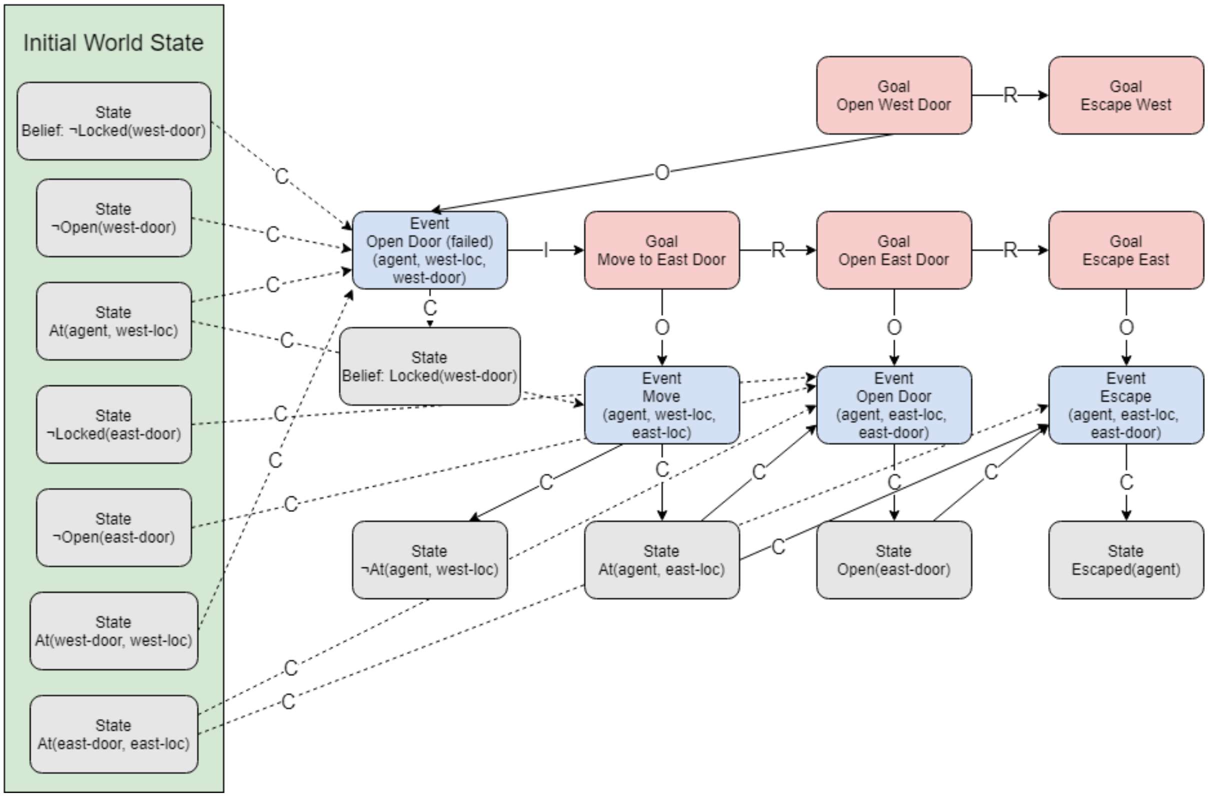

Fig. 3. The QKS generated for the escape scenario. As in Figure 1, arcs labeled with "R" are reason arcs, "O" are outcome arcs, "C" are consequence arcs, and "I" is the initiate arc. The initial states are included to provide consequence arcs for causal links relying on the initial state of the domain and are shown in dotted lines solely to aid in making the diagram easier to read.

only arcs of relevance are the forward reason, consequence, outcome, and initiate arcs. This allows the QKS to generate sets of answers for questions like:

- *What are the consequences of the character failing to open the west door?* The character wants to move to the east door because they fail to open the west door.

5 DISCUSSION AND FUTURE WORK

In this paper, we described the ways that QUEST models failed actions within its knowledge representation. We also defined one approach to translating narrative plans that contain failed actions into a QKS, which can then be used as a model for question answering in the context of computer generated stories. Our proposed translation algorithm to generate QKSs is ultimately intended to be used to experimentally evaluate user experiences of computationally generated narratives which incorporate failed actions. Our approach has not yet been validated and an evaluation is beyond the scope of this paper. In this section, we discuss a possible experimental evaluation to gauge effectiveness of this algorithm and its generated QKSs from plans with failed actions. We also discuss potential applications of this work.

5.1 Incorporating the Translation Algorithm into Experimental Evaluation

While this paper does not evaluate the efficacy of the proposed algorithm, this section outlines a possible experimental evaluation strategy. One approach for experimental evaluation would be to address how well a QKS generated from a plan matches the user's mental model of the plan formed after reading the plan's translation into narrative text, similar in fashion to the experimental evaluation by Riedl and Young [23]. As mentioned in previous sections of this paper, the QKS lends itself well as a model of question answering; the effectiveness of candidate answers to questions such as why, how, and what are the consequences can be evaluated by a QKS. Because of the knowledge required to understand how a QKS operates, we are unable to ask representative users to directly interpret the QKS and match it with their mental model. In fact, doing such would probably introduce significant bias. Instead, we leverage the question answering model of the QKS by presenting participants with question-answer pairs generated for the story and comparing their ratings of those pairs against the QKS's ratings.

The first step of our experiment would be to generate a story plan. The plan would need to be complex enough to give us confidence that the system works outside of simple examples, but it would also need to be short enough that the evaluation could proceed in a timely fashion. This story plan would then need to be converted into a story for participants to be able to understand it. We intend to use a simple template-based translation of the events that occurred in the plan, objectively translating these events into natural language. This approach may result in a story lacking engaging discourse structure, but this allows for strictly evaluating story event comprehension and reduces potential impact of bias introduced by hand-coded translation methods.

The proposed algorithm will take the story plan and produce a corresponding QKS. This QKS can be used to generate question-answer pairs for participants to respond to. Arc search procedures described by QUEST can be used to calculate arc distance. For example, we can follow a reason arc forward from one goal to another to generate the answer to a why question: "the agent wants to open the West door to escape to the West". Determining the question to be answered is a matter of taking the starting point of the arc and asking the question that matches the arc search procedure used: "why does the agent want to open the West door?". We would use a comprehensive algorithm that extracted all possible questions that could have answers simulated by our QKS. We would use the QKS to provide a set of valid answers and we would also randomly generate a set of invalid answers to the questions.

At this point, we would have a plan that has been objectively translated into a readable story, a QKS that is associated with that plan, an exhaustive set of questions that the QKS is capable of simulating answers for, and valid as well as invalid answers to those questions. The next step would be to give participants the story, present them with a question-answer pair, and determine how well their ratings of the pairs match the ratings generated by the QKS. There is a notion of arc length correlating with the strength of the QKS's simulated answers, so we would naturally expect lower ratings or even less accurate ratings from participants as the answer strength decreased in the QKS. In order for the QKS to be validated, participant ratings of the question-answer pairs (provided on a Likert scale, consistent with experimental methods used by Graesser and his colleagues [16]) must correlate with the QKS's ratings. We would expect question-answer pairs that were selected randomly with no valid arc connections in the QKS to be rated as bad answers while those with valid arc connections to be rated as good or very good depending on the strength of the same pair as estimated by the QKS.

5.2 Potential Applications

Cognitive model translation algorithms such as the QKS translation discussed in this paper are useful for a variety of efforts. Assuming the QKS generation method described here is validated, the following are a few valuable ways to apply it for future research.

One way to make use of the ability to relate narrative plan structures to the cognitive models of a narrative's readers or viewers is to examine the ways that choices around *narrative discourse* impact a user's comprehension. A narrative's discourse [8] is the realization of its communicative elements (e.g., its text, cinematic shots/shot sequences, narration), and decisions made by an author around discourse content impact the trajectory of a narrative consumer's experience (e.g., [2, 9, 31]). Given a particular story plan and the associated QKS, the impact of different discourse choices could be evaluated by measuring how those choices impacted a reader's mental model. This can be especially important in stories with failed actions, where comprehension might be limited without a discourse conveying the intentions behind the attempted action or the reasons that it failed.

Further, just as we added failed actions to the QKS here, additional work could build on our algorithm to translate story plans with greater expressive range. A new translation model could be defined that maps plan structures into other cognitive models such as the event indexing model [33] or the event horizon model [21]. Continuing to expand the functionality of cognitive model translation algorithms would allow them to be generated for a larger space of narratives by incorporating more narrative features, making them more broadly useful for evaluating and analyzing a wide variety of different narratives.

Finally, an author who wanted to use a narrative generation system could benefit from the accessibility of the QKS's question answering model. The author may not know or understand the internal algorithms or structures used by the system, so after defining a narrative domain and problem and having the planner create a solution plan, they might have trouble understanding the result. A QKS translation algorithm could generate a QKS capable of simulating the answers to questions that allow the author to understand what caused the final plan. For example, if an agent attempted to perform an action that failed, the author could ask the QKS why the agent was attempting to perform that action to inspect the agent's intention plan. By asking about the consequences of a failed action, they could understand the agent's new intention plan that was adopted once the action failed and the agent re-planned. Additionally, the author could ask how the action failed and could understand the sequence of actions which enabled the preconditions of the failing action, and after understanding this sequence, they could modify the narrative problem to avoid undesired events. In this way, a QKS can facilitate a non-expert user's interaction with narrative generation systems to ensure they generate the experience desired by the author.

6 CONCLUSION

As the expressive range of narrative generation systems increases, existing evaluation methods can be expanded to account for novel plot/plan structures with minimal impact on previous well-founded experimental designs. This paper highlights previously unused aspects of the QUEST cognitive model that align with novel features of stories (i.e., failed actions) being generated in recently developed planning systems. We define a translation algorithm that takes as input a story plan containing failed actions and creates a knowledge structure consistent with QUEST's QKS definitions. These knowledge structures can then be integrated into human subjects experiments comparing user experience during plan comprehension with the plan structures service as the plot line's source. Our intent is to use this translation algorithm in a study gauging the effectiveness of HeadSpace [30] plans to prompt specific mental models of a story.

REFERENCES

[1] Ruth S Aylett, Sandy Louchart, Joao Dias, Ana Paiva, and Marco Vala. 2005. FearNot!–an experiment in emergent narrative. In *International Workshop on Intelligent Virtual Agents*. Springer, Berlin, Heidelberg, 305–316.

[2] Byung-Chul Bae and R.M. Young. 2010. Suspense? Surprise! or How to Generate a Story with Surprise Endings Using a Disparity of Knowledge between the Reader and Characters in the Story. In *Proceedings of the International Conference on Interactive Digital Storytelling*. Springer, Berlin, Heidelberg, 304–307.

[3] Julio César Bahamón, Camille Barot, and R Michael Young. 2015. A Goal-Based Model of Personality for Planning-Based Narrative Generation.. In *AAAI*. AAAI, Palo Alto, California, USA, 4142–4143.

[4] Julio César Bahamón and R Michael Young. 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*.

[5] Jerome Bruner. 1987. Life as narrative. *Social research* (1987), 11–32.

[6] Rogelio E Cardona-Rivera, Thomason Price, D Winer, and R Michael Young. 2016. Question answering in the context of stories generated by computers. *Advances in Cognitive Systems* 4 (2016).

[7] Marc Cavazza, Jean-Luc Lugrin, David Pizzi, and Fred Charles. 2007. Madame bovary on the holodeck: immersive interactive storytelling. In *Proceedings of the 15th ACM international conference on Multimedia*. 651–660.

[8] David Chapman. 1987. Planning for Conjunctive Goals. *Artificial Intelligence* 32, 3 (1987), 333–378.

[9] Y. Cheong and R.M. Young. 2006. A Computational Model of Narrative Generation for Suspense. In *Working Notes of the AAAI 2006 Workshop on Computational Aesthetics*. 8–14.

[10] Yun-Gyung Cheong and R Michael Young. 2014. Suspenser: A story generation system for suspense. *IEEE Transactions on Computational Intelligence and AI in Games* 7, 1 (2014), 39–52.

[11] David Christian and R. Michael Young. 2004. Comparing cognitive and computational models of narrative structure. In *Proceedings of the National Conference on Artificial Intelligence*. American Association of Artificial Intelligence, AAAI, Menlo Park, CA, 385–390.

[12] Pablo Delatorre, Carlos León, Pablo Gervás, and Manuel Palomo-Duarte. 2017. A computational model of the cognitive impact of decorative elements on the perception of suspense. *Connection Science* 29, 4 (2017), 295–331.

[13] Joao Dias and Ana Paiva. 2005. Feeling and reasoning: A computational model for emotional characters. In *Portuguese conference on artificial intelligence*. Springer, 127–140.

[14] Maria Fox and Derek Long. 2003. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research* 20 (2003), 61–124.

[15] Arthur Graesser, Jonathan M Golding, and Debra L Long. 1991. Narrative representation and comprehension. (1991).

[16] Arthur C Graesser and Leslie F Clark. 1985. *Structures and procedures of implicit knowledge*. Vol. 17. Ablex Publishing Corporation.

[17] Arthur C Graesser, Sallie E Gordon, and Lawrence E Brainerd. 1992. QUEST: A model of question answering. *Computers & Mathematics with Applications* 23, 6-9 (1992), 733–745.

[18] Michael Mateas and Andrew Stern. 2003. Façade: An experiment in building a fully-realized interactive drama. In *Game developers conference*, Vol. 2. 4–8.

[19] Brian O'Neill and Mark Riedl. 2014. Dramatis: A Computational Model of Suspense. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, Québec, Canada) *(AAAI'14)*. AAAI Press, 944–950. http://dl.acm.org/citation.cfm?id=2893873.2894021

[20] Julie Porteous and Alan Lindsay. 2019. Protagonist vs antagonist provant: Narrative generation as counter planning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 1069–1077.

[21] Gabriel A. Radvansky and Jeffrey M. Zacks. 2017. Event Boundaries in Memory and Cognition. *Current Opinion in Behavioral Sciences* 17 (2017), 133–140.

[22] M Riedl. 2004. Narrative Generation: Balancing Plot and Character (Doctoral dissertation). *North Carolina State University, Raleigh, NC* (2004).

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

[24] Christian Roth and Hartmut Koenitz. 2019. Bandersnatch, yea or nay? Reception and user experience of an interactive digital narrative video. In *Proceedings of the 2019 ACM International Conference on Interactive Experiences for TV and Online Video*. 247–254.

[25] Alireza Shirvani, Stephen G. Ware, and Rachelyn Farrell. 2017. A possible worlds model of belief for state-space narrative planning. In *Proceedings of the 13th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*. 101–107.

[26] Gillian Smith and Jim Whitehead. 2010. Analyzing the expressive range of a level generator. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. 1–7.

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

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

[29] B. Thorne and R. M. Young. 2017. Generating Stories that Include Failed Actions by Modeling False Character Beliefs. In *Working Notes of the AIIDE Workshop on Intelligent Narrative Technologies*. AAAI, Salt Lake City, UT.

[30] Brandon R Thorne and R Michael Young. 2017. Generating stories that include failed actions by modeling false character beliefs. In *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*. AAAI, Palo Alto, California, USA.

[31] Tom Trabasso and Linda L Sperry. 1985. Causal relatedness and importance of story events. *Journal of Memory and language* 24, 5 (1985), 595–611.

[32] Stephen G Ware and Robert Michael Young. 2011. CPOCL: A Narrative Planner Supporting Conflict.. In *Proceedings of the Conference on AI and Interactive Digital Entertainment*.

[33] Rolf A. Zwaan, Mark C. Langston, and Arthur C. Graesser. 1995. The Construction of Situation Models in Narrative Comprehension: An Event-Indexing Model. *Psychological Science* 6, 5 (1995), 292–297.