Discourse-Driven Narrative Generation with Bipartite Planning

David R. Winer and **R. Michael Young**
Department of Computer Science
North Carolina State University
drwiner@ncsu.edu, rmyoung@ncsu.edu

Abstract

During content planning, a typical discourse generation system receives as input a library of facts and selects facts to include as the content for utterances. However, storytellers do not need to be completely constrained by a set of facts and instead can invent facts which support the storytellers goals, subsequently constructing the storyworld around those facts. We present a discourse-driven approach to narrative generation leveraging automated planning which can interleave construction of story and discourse while preserving modularity.

Introduction

Artificial intelligence (AI) automated planning research (Ghallab et al., 2004) is a popular source of data structures and algorithms for understanding, generating, and reasoning about stories (Young et al., 2014). Narratologists frequently distinguish the fabula (i.e., story) of narrative from the discourse (e.g., the narration of the story to a spectator) (Genette and Lewin, 1983; Bruner, 1991; Herman, 2013), and plans have proven useful for modeling both story and discourse (Young, 2007); they are effective for modeling discourse because a coherent sequence of communicative actions is plan-like (Cohen and Perrault, 1979; Lambert and Carberry, 1991; Young and Moore, 1994), and plans are effective for modeling stories because stories are composed of events with cause-effect relations and characters also form plans to achieve their goals (Trabasso and Sperry, 1985; Riedl and Young, 2010). Behavioral research demonstrates that plans capture many key features that spectators use to understand narrative discourse (Trabasso and Sperry, 1985; Christian and Young, 2004; Ware et al., 2014; Radvansky et al., 2014; Cardona-Rivera et al., 2016).

A typical approach to discourse generation is to supply a program with a library of facts about the domain of interest from which to select content for utterances (Meteer, 1991; Reiter and Dale, 1997). *Narrative* discourse generation systems are usually no different (Lönneker, 2005; Callaway and Lester, 2002); story is generated to meet some user-provided goals (i.e., a *story plan* solves a *story problem*) and passed through a pipeline as input for generating discourse and narration (e.g., text or animation) to solve a *discourse problem* (Callaway and Lester, 2002; Young et al., 2004; Jhala and Young, 2010; Cheong and Young, 2015) (see also (Young et al., 2014)). This pipeline architecture is amenable to the task of generating different discourse plans about the same set of events.

However, analysis of these programs reveals that if there are story constraints associated with discourse planning operations, an input plan that solves a story problem may not meet those constraints needed to solve the discourse problem (i.e., the story plan is *incompatible* with the discourse goals), even though a compatible solution to the story problem exists. For example, to tell stories about characters who courageously navigate a dangerous terrain, one discourse action might be to convey that an obstacle, such as a bridge, is dangerous and has the constraint that some character dies at

Proceedings of The 9th International Natural Language Generation conference, pages 11–20,
Edinburgh, UK, September 5-8 2016.

this obstacle; if the discourse planner receives as input a story where no character dies, the discourse planner would be unable to meet the discourse goal or subgoal to convey an obstacle is dangerous, even though in the space of possible stories, there exists at least one story where at least one character dies. The discourse plan may depend on describing the obstacle as dangerous as a causal antecedent for conveying that the main protagonist is in danger because the protagonist is at the obstacle. A planner which generates both story and discourse plans from story and discourse problems is considered *bipartite complete* if the planner will find a compatible pair of story and discourse solutions if one exists.

Storytellers needn't be completely limited by existing facts and can invent facts to support their storytelling goals, subsequently building the storyworld around those facts. For example, a screenwriter may add an event to the story (e.g., a non-central character slips to his death from a narrow bridge), in order to elicit a discourse effect (e.g., believing the bridge is a dangerous obstacle for the protagonist). Following this idea, our approach to narrative generation is discourse-driven; as a discourse planner constructs a discourse plan, constraints on the scenarios in the story are added to the story plan so that a story planner constructs a story plan which is compatible with the discourse plan. Our representation of planning formalisms enables constraints to be *flexible* (they need not completely specify the underlying scenario), *narrative theoretic* (they can refer to high-level phenomena such as character intentionality and conflict), and to be *path pruning* (they can speed up the search for a solution by limiting exploration to just those plans which are consistent with the constraints).

In most classic NLG systems, generation of both the story and discourse would fall under the task of content determination (Reiter and Dale, 1997). However, our criticism of the story-then-discourse pipeline architecture is reminiscent of past discussions on the drawbacks of using a modular pipeline in NLG systems (Reiter, 2000) such as the well-known ”generation gap” between text planning and realization (Meteer, 1991). We introduce *bipartite planning* which preserves modularity of story and discourse, allowing users to plug in different story and discourse problems and potentially different generation systems, but which interleaves construction of story and discourse to enable bipartite completeness.

Related Work

Narrative planning benefits from a rich history of AI research. The first story generation system to use planning is TALESPIN (Meehan, 1977), which generates stories about woodland creatures that take actions to satisfy simple needs in accordance with rules of the world. Another early story generation system UNIVERSE (Lebowitz, 1983) represents plot fragments as plans and selects a fragment to execute if it satisfies an authorial goal. MINSTREL (Turner, 1993) uses planning to create an outline of a story and case-based reasoning to fill in details from a story library. Cavazza and colleagues use forward state-space character-centric planning but cannot guarantee authorial goal conditions are achieved (Cavazza et al., 2002; Porteous et al., 2010). Plan-space search is used as a top-down approach to story generation; a user specifies initial and goal states of a story world and the solution space is restricted to just those story plans which are causally sound (Young et al., 2014), and narrative-theoretic extensions (e.g., IPOCL (Riedl and Young, 2010) and CPOCL (Ware et al., 2014)) further limit the solution space just to plans where characters act believably.

Narrative discourse generation systems (NDGS) typically take story as input from a library of facts such as from a story planner or from collected data and produce a plan for narrating the story. STORYBOOK (Callaway and Lester, 2002) is an end-to-end narrative prose generation system with four parts: 1) a narrative organizer which takes as input a story plan, 2) a sentence planner which creates a proto-sentence outline, 3) a revision module which produces prose paragraphs from proto-sentences, and 4) a surface realizer which makes some grammatical edits and formats the story in a file. However, STORYBOOK expects decisions about discourse, such as the order to tell events or how to describe elements of the story, such as from a particular character perspective (focalization), to be provided as part of the input (in the story plan). A distinct discourse module for content determination has been proposed as an amendment to this architecture (Lönneker, 2005). This more closely mirrors the way narratologists partition narrative into story and discourse (Genette and Lewin, 1983; Young, 2007).

Later NDGSs have included a discourse content determination module which structures events in story plans provided as input. Suspenser (Cheong and Young, 2015) and Prevoyant (Bae and Young, 2014) systems arrange the events in a story into an ordering which elicits suspense or surprise, respectively, based on cognitive-computational definitions inspired by psychological theory). These systems first determine which events from the story are worth telling by measuring their causal importance (Young, 1999), and include less important events when they can help maximize a suspense/surprise function by increasing the distance between important events. Darshak (Jhala and Young, 2010) is a cinematic discourse generation system in which dramatic patterns decompose into camera shot patterns to convey events from an input story plan which are animated in a virtual environment. These systems use a discourse planner adapted from the DPOCL (decompositional) algorithm (Young et al., 1994).

One drawback about these systems is that the communicative actions are all *inform* speech acts (suspense and surprise decompose into inform actions) and lack representation for evaluative description (e.g., a *brave* character, an *honorable* action, a *dangerous* location, etc.). An interesting approach by Li, Thakker, Wang, and Riedl (2014) is to learn the most probable sentiment of a story word from a corpus of annotated words and from crowdsourced stories involving a similar scenario to the story. The words used in the text can then be replaced to make the story more descriptive and to reflect the mood of the scene. Our vision of discourse generation more closely aligns with the properties described by Grosz and Sidner (1986): discourse reasoning ought to be goal-oriented, such that descriptions causally enable subsequent descriptions about the story. Bae, Cheong, and Young (2011) produce narrative plans with focalization by using different plan libraries for different characters which have built-in descriptions of the story based on the character's persona. The events are reconstructed with the plan library representing the character who's perspective the story is being told from. However, tailoring these descriptions for a particular story is time consuming, and a modular, domain-independent approach is preferred where a user can swap out description knowledge and tell the same kind of narrative with different storyworlds and characters.

Problem Formulation

Our discourse-driven narrative planning approach is a search for solutions to two problems, a story problem and discourse problem, where a solution is a plan of actions to bring an initial state to a goal state. At the story level, the solution represents the actions of characters in the storyworld, whereas at the discourse level, the solution represents the communicative actions by a narrator agent to inform and describe elements in the story to a spectator agent. A pair of compatible story and discourse solutions to story and discourse problems is a *bipartite solution*. With prior approaches to story and discourse generation, a story solution is supplied as input to a discourse planner. In this approach, elements in a story solution become required as part of the search for a discourse solution. These requirements are prerequisite criteria about the story for the story and discourse plans to be a bipartite solution to the story and discourse problems.

Narrative Plans

Partial-order causal link (POCL) planning is a type of *planning as refinement search* (Kambhampati et al., 1995), involving search through plan-space such that each child node in the search is a refinement to the (potentially flawed) plan represented at its parent node. Through an iterative process of identifying flaws in the plan and repairing them in a least-commitment manner (Weld, 1994), plans with no flaws are selected and returned as solutions to the planning task. A **planning problem** or task consists of an initial state, a set of goal conditions, a set of action types, and a set of logical **constants**; in a storyworld, constants are characters, items, and locations, whereas at the discourse level, constants refer to elements of the story (e.g. a character, a character's plan, a conflict, etc.).

Figure 1 Example story action operators and planning problem

```
              Miniature Story Planning Problem

"move" story operator
variables = {                       Constants: Indy, Sapito,
      ?c - character                cliff1, cliff2, bridge
      ?from ?to - location}         Operators:{
precond(move, (at ?c ?from))           move,
precond(move, (adj ?from ?to))         fall-from}
precond(move, (alive ?c))           Initial: {
effect(move, (at ?c ?to))              (at Indy cliff1),
effect(move, ¬(at ?c ?from))           (at Sapito cliff1),
                                       (adj cliff1 bridge),
"fall-from" story operator             (adj bridge cliff2),
variables = {                          (intends Sapito
      ?c - character                      (at Sapito cliff2)),
      ?from - location}                (intends Indy
precond(fall-from, (at ?c ?from))         (at Indy cliff2))}
precond(fall-from, (high-up ?from)) Goal: {
precond(fall-from, (alive ?c))         (at Indy cliff2),
effect(fall-from, ¬(alive ?c))         (alive Indy)}
effect(fall-from, ¬(at ?c ?from))
```

Action Operators

STRIPS-style **operators** (Fikes and Nilsson, 1972) depict action types that can occur, and a **step** is an instance of an action operator. Each operator has **preconditions**, describing what must be true for the step to occur, and **effects**, representing how the world will change after the step occurs. Preconditions and effects are described in a language of function-free first-order predicate literals. These literals have variables which must be assigned to constants in the planning problem. For instance, consider a story action operator type ”fall-from” which may be found in a story problem (see Figure 1). The operator has variables ?character and ?from, preconditions that ?character is at ?from, ?character is alive, and that ?from is high-up, and effects that ?character is not at ?from and is not alive, reflecting the change in the world after the action executes. Figure 1 shows two example action operators written in a format amenable for understanding operations we discuss later.

At the discourse level, the world state is a conjunction of literals indicating what a spectator agent believes is true and not true about the story (denoted with *bel* before the believed literal). Discourse actions are communicative actions taken by a narrator agent to add or remove the spectator’s beliefs. In addition to preconditions and effects, discourse action operators have **requirements** and **restrictions** on the variables used in its preconditions and effects. A discourse action cannot be used if one of its set of restrictions are detected in the story, and the story plan is considered incompatible if any set of restrictions from discourse actions are detected during its construction. When a discourse action is used as a step in a discourse plan, the requirements are matched to existing elements of the story plan and/or are added to the story plan.

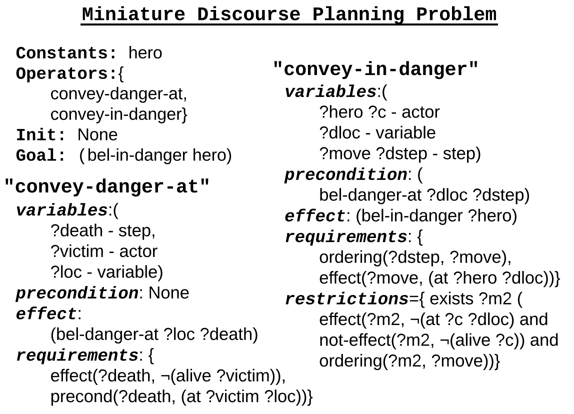
When the variable in a requirement is a step in the story plan, the step might only be partially defined, called a **partial step**. The planner must select some action operator which is consistent with the step and its requirements and add components to the partial step so that the step is an instance of the operator. For example, a discourse action to convey that a location is dangerous (see Figure 2) has the requirements that some story action occurs with the precondition that a character is at this location and the effect that this character is not alive. The fall-from action operator (see Figure 1) would be considered consistent with these requirements. The rest of this step would be added to the story plan such as the preconditions that the location is high-up and the character is alive (see Figure 3). At some point, the variable representing the high-up location would be assigned to a constant from the story problem with the desired property, such as the bridge. Figure 2 shows two example discourse operators including the example discussed here.

The variable in a discourse action can have a type associated with any element of a story plan, such as a variable, step, set of steps, ordering of steps, causal link between steps, character plan, character goal, etc. This enables discourse actions to post requirements and restrictions about many aspects of the story and thus enables a wide range of descriptions to be readily supported in discourse actions. Each discourse action could also include communicative actions (e.g., camera shots, text operators, etc) that are designed by artists/writers to narrate the story content, similar to the design in DARSHAK (Jhala and Young, 2010) in which the communicative effects of a scene are decomposed into primitive camera shots.

Causal Soundness

Each plan has one placeholder **start step** whose effects express the initial state, and one placeholder **end step** whose only preconditions are the goal conditions that must be true at the end of the plan. All steps in a solution are goal-oriented; each step’s

Figure 2 Example discourse action operators and planning problem



```
                Miniature Discourse Planning Problem

  Constants: hero
  Operators:{                          "convey-in-danger"
       convey-danger-at,                variables:(
       convey-in-danger}                     ?hero ?c - actor
  Init: None                                 ?dloc - variable
  Goal: (bel-in-danger hero)                 ?move ?dstep - step)
                                        precondition: (
"convey-danger-at"                           bel-danger-at ?dloc ?dstep)
  variables:(                           effect: (bel-in-danger ?hero)
       ?death - step,                   requirements: {
       ?victim - actor                       ordering(?dstep, ?move),
       ?loc - variable)                      effect(?move, (at ?hero ?dloc))}
  precondition: None                    restrictions={ exists ?m2 (
  effect:                                    effect(?m2, ¬(at ?c ?dloc) and
       (bel-danger-at ?loc ?death)           not-effect(?m2, ¬(alive ?c)) and
  requirements: {                            ordering(?m2, ?move))}
       effect(?death, ¬(alive ?victim)),
       precond(?death, (at ?victim ?loc))}
```

preconditions are causally linked to one or more effects from prior steps. A **causal link** between steps s and t, denoted $s \xrightarrow{p} t$, indicates that s has an effect p which co-designates with a precondition p of step t. Step s is an *ancestor* of step t, and t is a *descendant* of step s. A step's *causal ancestors* are all steps in the transitive closure of the ancestor relationship. A step's *causal descendants* are all steps in the transitive closure of the descendant relationship.

A precondition which is not yet make true by another step (i.e., an **open precondition flaw**) is resolved by finding an action operator which has an effect that can become the needed precondition. When a step can possibly undo one of the effects of a prior step which is needed as a precondition of a later step (i.e., a **threatened causal link flaw**), the planner attempts to reorder steps or add constraints to the steps involved so that no conflict can arise. The plan is *causally sound* just when for every total ordering of steps, each step's preconditions are met when that step is executed.

Intentional Coherence

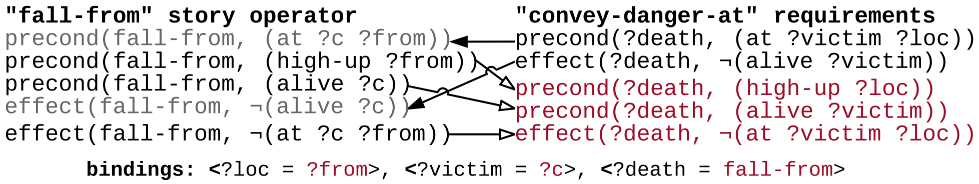
Another requirement adopted for story plans is that each character should only intentionally take actions which can be explained as part of a character plan to achieve one of that character's goals. An **intention frame** includes the elements needed to represent why a character adopts a goal and her actions to achieve it. It is a tuple $\langle a, g, m, s_f, S_f \rangle$ where a is an actor, g is some literal that a wishes to make true, m is a motivating step whose effects include $intends(a, g)$, s_f is the satisfying step whose effects include g, and S_f is a subplan for a to satisfy g such that all steps in S_f are causal ancestors of s_f and have the consent of a. An action which does not require a "volunteer", such as an accident or a force of nature, is called a **happening**. The solution is considered *intentionally coherent* just when when all voluntary actions are part of a character's intention frame. Characters do not always achieve their goals and may take the first n steps of the subplan. For example, a character a may not finish a plan because some other step undoes one of the effects of a step that a took to enable a subsequent step. The different ways that a character's plan can become thwarted by another step has been studied in prior work as a definition of narrative conflict (Ware et al., 2014). In story problems supporting character intentionality, characters either have goals in the initial state or adopt goals as the effects of actions taken in the story (e.g., a character who finds a treasure map might adopt the goal to have the treasure).

Solution

To solve the planning problems (one story problem and one discourse problem), the planner iteratively selects flaws from either plan and selects some method to resolve the flaw. The solutions are plans: a plan is a tuple $\langle S, B, O, L \rangle$ where S is a set of steps, B is a set of bindings between variables in S, O is a set of ordering constraints overs steps in S, and L is a set of causal links between steps in S. The plan is *valid* just when the plan is causally sound and all constants have an assigned variable. Additionally, story plans include a set of character intention frames. The story plan is *valid* just when the base plan is valid plus the plan is intentionally coherent and all variables are assigned to a constant[1]. The two plans are structurally similar but conceptually distinct. In the story plan, the steps are actions taken by characters. In the discourse plan, steps are communicative actions taken by a narrator agent to add or remove the spectator's beliefs. Once a compatible pair of solutions are constructed, the

[1] Stories are limited by the constants provided as input, whereas discourse variables may refer to elements created during story planning

Figure 3 Partial story step (and discourse variable) ?death in the discourse action ”convey-danger-at” becomes an instance of the ”fall-from” operator in the story plan. The solid arrows indicate which requirements are matched to elements of ”fall-from”. The hollow arrows indicate which elements of the operator are added to the requirements, so that ?death becomes an instance of ”fall-from”.



```
"fall-from" story operator                "convey-danger-at" requirements
precond(fall-from, (at ?c ?from))         precond(?death, (at ?victim ?loc))
precond(fall-from, (high-up ?from))       effect(?death, ¬(alive ?victim))
precond(fall-from, (alive ?c))            precond(?death, (high-up ?loc))
effect(fall-from, ¬(alive ?c))            precond(?death, (alive ?victim))
effect(fall-from, ¬(at ?c ?from))         effect(?death, ¬(at ?victim ?loc))

          bindings: <?loc = ?from>, <?victim = ?c>, <?death = fall-from>
```

solutions can be sent down the NLG pipeline to be realized as text or some in some other medium. For instance, the story could be used to animate avatars in a virtual world and the discourse actions could be mapped to camera actions to film the events.

Algorithm

The bipartite planning algorithm **BiPOCL** for generating story and discourse solutions to story and discourse problems is presented in Algorithm 1. In discourse planning (lines 3-8), flaws are added for requirements needed in the story plan. Story planning (lines 9-14) involves a combination of approaches from prior work (Riedl and Young, 2010; Ware, 2014) which are not provided in detail here (such as those involving intention frames). In addition, story planning involves selecting requirements to add to the story plan and partial steps in the story plan to make into instances of story action operators. Threats to causal links (lines 15-19) are resolved, with the exception that some threats are okay to leave in the story plan if they can represent conflict. The algorithm terminates when bindings or orderings are inconsistent in either plan, or when there are no flaws in either plan.

Example: The Dangerous Bridge

To help explain the bipartite planning approach to narrative generation, a miniature example is presented consisting of story and discourse problems and a bipartite solution. The story domain and problem for this example is inspired by Indiana Jones (see Figure 1). Agents in this domain can move between adjacent locations and fall by accident from high locations. The discourse problem contains actions for describing story elements as dangerous (see Figure 2). The discourse action ”convey-danger-at” describes a location as dangerous by virtue that some character dies as a consequence of being at this location. The discourse action ”convey-in-danger” describes a character as being in danger by virtue that this character is at a dangerous location (precondition, requirement), and no other character before this has safely moved from this location and left without dying (restriction). Both problems are provided as input by users.

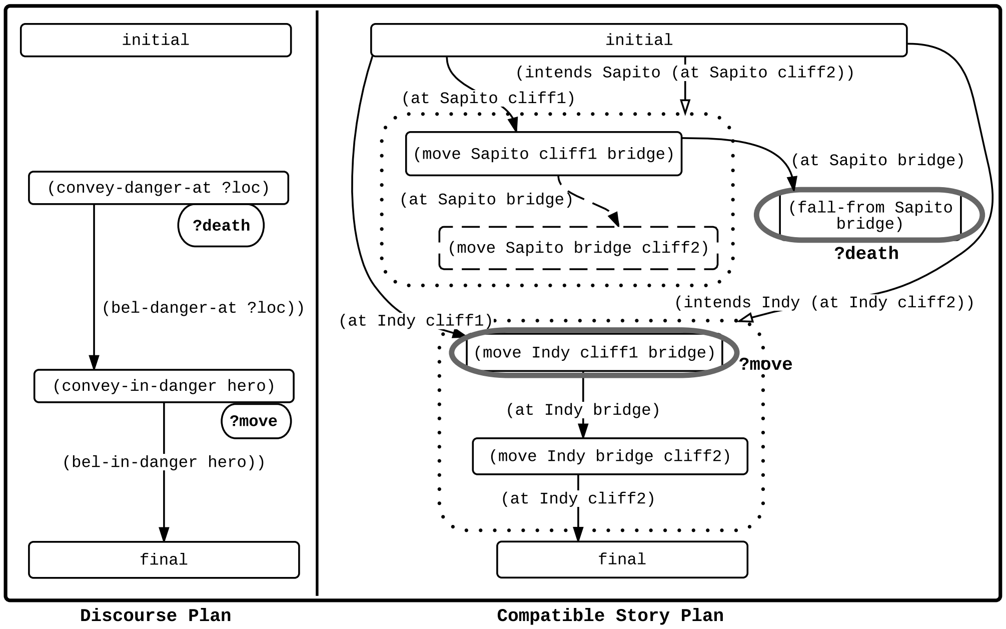
An instance of a bipartite solution to the story and discourse problems is presented in Figure 4. Space limitations prevent us from including figures to demonstrate the construction of this solution. For details on the construction of intention frames, see (Ware, 2014). To start, the BiPOCL is called with the initial plans, each containing a start and end step. In the story plan, there are two empty intention frames motivated by the start step which must explain how Indy and Sapito will achieve their goal to be across the bridge at cliff2. Open precondition flaws are added for every goal condition of both plans. Flaws can be selected in any order.

The open precondition `(at Indy cliff2)` is repaired by adding a causal link from a new step `move Indy bridge cliff2` which has the new open precondition `(at Indy bridge)` (the bridge is the only location adjacent to cliff2). The open precondition that `(bel-in-danger hero)` is repaired by adding a causal link from a new step `convey-in-danger hero` which has new open precondition `(bel-danger-at ?dloc)`. The requirements for this discourse action can be added to the story immediately or at some later iteration. The story action `move Indy bridge`

Algorithm 1 The BiPOCL (Bipartite Partial Order Causal Link) Algorithm

```
Termination: If either plan is inconsistent, backtrack. Otherwise, return the plans.
Plan Refinement: Non-deterministically select a flaw in either plan
    Discourse Planning:
        Choose a precondition of a step not yet established through a causal link and either:
            Reuse: Find a step which already establishes the precondition.
            New: Create the step from an operator which establishes the precondition.
                Add flaws for the step's requirements which need to be added to the story.
        Add a causal link between the new/old step and the step with the precondition.
    Story Planning: Do one of the following:
        Choose a flaw in the story and resolve with associated refinement method.
        Choose a discourse requirement not yet added to the story, and either:
            Reuse: Find a story element which already satisfies the requirement.
            New: Create the requirement and add it to the plan.
        Choose a partial step, select a consistent action operator, and add the step's missing components.
Threat Resolution:
    Find a step which may threaten to undo a causal link. Choose how to prevent the threat:
        Promotion: If possible, move the threatened steps to occur before the threat in the plan.
        Demotion: If possible, move the threatened steps to occur after the threat in the plan.
        Restriction: If possible, add constraints to the steps involved so that no conflict can arise.
Recursive Invocation Call the planner recursively with the new plan structure.
```

Figure 4 On the left is a discourse plan; boxes are discourse steps, arrows are causal links, and variables of interest are in ovals. On the right is a compatible story plan; boxes are steps, a dashed arrow is a threatened causal link, a solid arrow is a causal link, a dotted bounding box is an intention frame, the source of a hollow arrow is a motivating step and the sink is an intention frame, the dashed box in an intention frame indicates a step not taken (but which completes the character's plan), and an oval surrounds an element bound to the labeled discourse variable.



cliff2 is a candidate for discourse variable ?move, making the dangerous location (?dloc) cliff2. The search would fail in this case because in our miniature universe, there exists no action that can bring death to a character at cliff2 (since cliff2 is not high-up). Alternatively, a new partial step could be added to the story plan, leaving the location yet unspecified. The partial step is only consistent with the ”move” operator, creating step instance move ?c ?from ?to which floats in the plan ordered between the start and end steps. The open precondition (at Indy bridge) can be repaired by adding a causal link from this floating step, which becomes move Indy ?from bridge. The open precondition flaw for this step can be repaired by adding a causal link from the start step with effect (at Indy cliff1), transforming the required step into move Indy cliff1 bridge. Since this step is bound to the discourse variable ?move, the bridge is the dangerous location ?dloc.

The open precondition (bel-danger-at bridge) (since now ?dloc=bridge) is repaired by a causal link from new step convey-danger-at bridge. Its requirement is that ?death is a step ordered before ?move (via binding to variable ?dstep in convey-in-danger) with the effect that ?victim is not alive. Only Sapito is a valid candidate to be ?victim. The requirement can be fulfilled by adding step fall-from Sapito bridge. Sapito's plan to be at cliff2 can be constructed like Indy's. Either Sapito can cross the bridge to cliff2 (fulfilling his goal) and then move back to the bridge, or he can get to the bridge and fall there without ever completing the goal. There is a restriction about the bridge from action convey-in-danger that no character can move from the bridge alive if that move occurs before Indy's action of moving to the bridge. Thus, Sapito must fall from the bridge without reaching cliff2.

Some of the relationships between discourse variables and story elements are marked in the figure. The bindings between discourse variables and story elements include ⟨hero, Indy⟩, ⟨?victim, Sapito⟩, ⟨?loc, bridge⟩, ⟨?move, move Indy cliff1 bridge⟩, and ⟨?death, fall-from Sapito bridge⟩.

The resulting narrative is that Sapito tries to cross the bridge but accidentally falls and dies so that the spectator believes the bridge is dangerous. Then, Indy moves to the bridge so that we believe he is in danger. Finally Indy safely crosses the bridge and achieves his goal. Medium-specific realization depends on system goals and can be as simple as filling slots in templates associated with discourse actions.

Discussion and Future Work

Typically, NDGSs accept as input a set of propositions or a database of facts from which to select for utterances in a communicative plan and are not bipartite complete. Our discourse-driven approach is likely bipartite complete because requirements limit the solution space for stories during discourse plan refinement rather than limiting the solution space for discourse plans during discourse plan refinement, avoiding the possibility for inconsistencies between requirements and existing story elements.

We call our narrative generation approach *discourse-driven* because the propositions about the domain of facts (i.e., the story world and character actions) are created to support the storyteller's goals. This shift of responsibility is appropriate for storytelling where an author/narrator may not be reporting on real events and instead invents scenarios for her characters to elicit a desired effect. With our approach, a story generation system accommodates these scenarios in a causally sound and intentionally coherent world. We presented an example motivated by Indiana Jones where the narrative planner adds an event to the story level (i.e., a non-central character slips off a bridge), in order to construe the bridge as dangerous.

Prior approaches to discourse planning have focusing on what order to inform events to a spectator (presentation order), rather than on describing events in a goal-oriented way. Research on discourse comprehension demonstrates that causal reasoning is biased by the recency of narrative events in text, which may have implications for timing (Winer et al., 2015) and recallability of events (Cardona-Rivera et al., 2012). Bipartite planning could set up good timing through a specification of minimum distances between steps in a story that are relevant for the discourse plan, or by structuring parallel plot lines to enable juxtaposition editing.

The story-then-discourse approach may not be well suited for ”description planning” because it is unlikely that the propositions provided as input to the discourse planner contain the scenario or context needed for a particular description to be applicable. Our approach depends on the assumption that descriptions can have causal relationships, an assumption that ought to be evaluated empirically. In prior work, generated story plans have been automatically mapped to a psychological model of question-answering and shown to have representational accuracy on some key question types (Christian and Young, 2004; Cardona-Rivera et al., 2016). This model may be extended to include question types about a spectator's interpretation via information described at the discourse level. For example, if a story element is evidence for some belief (e.g., the death of Sapito is evidence that the bridge is dangerous), then our model may predict which elements of the story ought to be used to justify that belief (e.g., *Q*: ”When did you begin to believe that the bridge is dangerous?” *A*: ”When Sapito fell from the bridge.”).

References

Byung-Chull Bae and R Michael Young. 2014. A computational model of narrative generation for surprise arousal. *Computational Intelligence and AI in Games, IEEE Transactions on*, 6(2):131–143.

Byung-Chull Bae, Yun-Gyung Cheong, and R Michael Young. 2011. Automated story generation with multiple internal focalization. In *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, pages 211–218. IEEE.

Jerome Bruner. 1991. The narrative construction of reality. *Critical inquiry*, pages 1–21.

Charles B Callaway and James C Lester. 2002. Narrative prose generation. *Artificial Intelligence*, 139(2):213–252.

Rogelio E Cardona-Rivera, Bradley A Cassell, Stephen G Ware, and R Michael Young. 2012. Indexter: A computational model of the event-indexing situation model for characterizing narratives. In *The Workshop on Computational Models of Narrative at the Language Resources and Evaluation Conference*, pages 32–41.

Rogelio E Cardona-Rivera, Thomason Price, David R Winer, and R Michael Young. 2016. Question answering in the context of stories generated by computers. *Journal of Advances in Cognitive Systems*.

Marc Cavazza, Fred Charles, and Steven J Mead. 2002. Character-based interactive storytelling. *IEEE Intelligent systems*.

Yun-Gyung Cheong and R Michael Young. 2015. Suspenser: A story generation system for suspense. *Computational Intelligence and AI in Games, IEEE Transactions on*, 7(1):39–52.

David B Christian and R Michael Young. 2004. Comparing cognitive and computational models of narrative structure. In *AAAI*, pages 385–390.

Philip R Cohen and C Raymond Perrault. 1979. Elements of a plan-based theory of speech acts. *Cognitive science*, 3(3):177–212.

Richard E Fikes and Nils J Nilsson. 1972. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3):189–208.

Gérard Genette and Jane E Lewin. 1983. *Narrative discourse: An essay in method*. Cornell University Press.

Malik Ghallab, Dana Nau, and Paolo Traverso. 2004. *Automated planning: theory & practice*. Elsevier.

Barbara J Grosz and Candace L Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204.

David Herman. 2013. *Storytelling and the Sciences of Mind*. MIT press.

Arnav Jhala and R Michael Young. 2010. Cinematic visual discourse: Representation, generation, and evaluation. *Computational Intelligence and AI in Games, IEEE Transactions on*, 2(2):69–81.

Subbarao Kambhampati, Craig A Knoblock, and Qiang Yang. 1995. Planning as refinement search: A unified framework for evaluating design tradeoffs in partial-order planning. *Artificial Intelligence*, 76(1):167–238.

Lynn Lambert and Sandra Carberry. 1991. A tripartite plan-based model of dialogue. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 47–54. Association for Computational Linguistics.

Michael Lebowitz. 1983. Creating a story-telling universe. *Proceedings of the Eighth International Joint Conference on Artifiial intelligence*, 1:63–65.

Boyang Li, Mohini Thakkar, Yijie Wang, and Mark O Riedl. 2014. Data-driven alibi story telling for social believability. *Social Believability in Games*.

Birte Lönneker. 2005. Narratological knowledge for natural language generation. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG-05)*, pages 91–100. Citeseer.

James R Meehan. 1977. Tale-spin, an interactive program that writes stories. In *IJCAI*, volume 77, pages 91–98.

Marie W Meteer. 1991. Bridging the generation gap between text planning and linguistic realization. *Computational Intelligence*, 7(4):296–304.

Julie Porteous, Marc Cavazza, and Fred Charles. 2010. Applying planning to interactive storytelling: Narrative control using state constraints. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 1(2):10.

Gabriel A Radvansky, Andrea K Tamplin, Joseph Armendarez, and Alexis N Thompson. 2014. Different kinds of causality in event cognition. *Discourse Processes*, 51(7):601–618.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(01):57–87.

Ehud Reiter. 2000. Pipelines and size constraints. *Computational Linguistics*, 26(2):251–259.

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

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

Scott R Turner. 1993. Minstrel: a computer model of creativity and storytelling.

Stephen G Ware, R Michael Young, Brent Harrison, and David L Roberts. 2014. A computational model of plan-based narrative conflict at the fabula level. *Computational Intelligence and AI in Games, IEEE Transactions on*, 6(3):271–288.

Stephen G Ware. 2014. *A Plan-Based Model of Conflict for Narrative Reasoning and Generation*. North Carolina State University.

Daniel S Weld. 1994. An introduction to least commitment planning. *AI magazine*, 15(4):27.

David R. Winer, Adam A. Amos-Binks, Camille Barot, and R. Michael Young. 2015. Good Timing for Computational Models of Narrative Discourse. In *6th Workshop on Computational Models of Narrative (CMN 2015)*, volume 45, pages 152–156.

R Michael Young and Johanna D Moore. 1994. Dpocl: A principled approach to discourse planning. In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 13–20. Association for Computational Linguistics.

R Michael Young, Martha E Pollack, and Johanna D Moore. 1994. Decomposition and causality in partial-order planning. In *AIPS*, pages 188–194.

R Michael Young, Mark O Riedl, Mark Branly, Arnav Jhala, RJ Martin, and CJ Saretto. 2004. An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development*, 1(1):51–70.

R Michael Young, SG Ware, BA Cassell, and Justus Robertson. 2014. Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives. *SDV. Sprache und Datenverarbeitung*.

R Michael Young. 1999. Using grice's maxim of quantity to select the content of plan descriptions. *Artificial Intelligence*, 115(2):215–256.

R Michael Young. 2007. Story and discourse: A bipartite model of narrative generation in virtual worlds. *Interaction Studies*, 8(2):177–208.