Cinematic Visual Discourse: Representation, Generation, and Evaluation

Arnav Jhala, *Member, IEEE*, and R. Michael Young, *Senior Member, IEEE*

***Abstract*—In this paper, we present the design, implementation, and evaluation of an end-to-end camera planning system called Darshak. Darshak automatically constructs cinematic narrative discourse of a given story in a 3-D virtual environment. It utilizes a hierarchical partial-order causal link (POCL) planning algorithm to generate narrative plans that contain story events and camera directives for filming them. Dramatic situation patterns, commonly used by writers of fictional narratives, are formalized as communicative plan operators that provide a basis for structuring the cinematic content of the story's visualization. The dramatic patterns are realized through abstract communicative operators that represent operations on a viewer's beliefs about the story and its telling. Camera shot compositions and transitions are defined in this plan-based framework as execution primitives. Darshak's performance is evaluated through a novel user study based on techniques used to evaluate existing cognitive models of narrative comprehension. Initial study reveals significant effect of the choice of visualization strategies on measured viewer comprehension. It further shows significant effect of Darshak's choice of visualization strategy on comprehension.**

***Index Terms*—Discourse generation, machinima generation, planning.**

THE automatic generation and communication of narrative are long-standing research areas within artificial intelligence [1]–[4]. To date, much of the research on story generation has focused either on computational models of plot and the construction of story events or on the communication of a given plot through the textual medium. Problems central to the textual communication of a narrative have much in common with challenges found in the generation of other genres of natural language discourse, including the critical issues of content determination (what propositions should appear in the text) and ordering (how should the propositions describing a story be ordered so as to present a coherent story to a reader).

Text-based storytelling systems (e.g., [4]) typically build upon computational models of discourse generation in order to produce coherent narratives. While these approaches have been successful within the text medium, less attention has been focused on techniques for the creation of effective *cinematic* discourse—the creation of narrative told using a virtual camera operating within a 3-D environment. As we know from our experiences as film watchers, cinema is a powerful and effective medium for communicating narratives.

This paper presents computational techniques for automatically constructing camera shot sequences for telling a story. This is one element of cinematic visual discourse, i.e., communication of stories through a sequence of intentionally crafted camera shots, and is implemented in a system called Darshak.[1] This is accomplished by representing cinematic conventions as plan operators and developing a planning algorithm that constructs visual discourse plans containing cinematic actions. Camera shots are represented as primitive operators whose execution manipulates the beliefs of viewers about the story world.

The discourse planning algorithm constructs a schedule of camera shots for a given story and a set of communicative goals. Saliency, coherency, and temporal consistency are identified as the three desired properties of the algorithm. Saliency is guaranteed by inclusion of explicit binding constraints on the camera operators relating their execution to story world entities. Temporal consistency is obtained through an explicit representation of temporal variables and a temporal consistency algorithm, added as an extension to a conventional discourse planning algorithm. Coherency is indirectly evaluated through cognitive metrics of question answering.

Evaluation of intelligent camera control systems is a challenging problem [5], primarily because there are several dimensions across which camera systems can be evaluated. Most current camera systems are evaluated based on their performance in terms of speed of calculating camera positions. While it is difficult to evaluate stylistic capabilities of such systems, it is possible to evaluate their efficacy in communicating the underlying narrative content. For this, an experiment was designed for comparing the effectiveness of different visualization strategies in communicating a story. This design is based on established cognitive models of story understanding [6] that have been successfully used to evaluate plan-based computational models of narrative [7]. Initial experiments establish that visualization strategy significantly affects viewer comprehension, and intentionally generated visualizations by the system result in improved comprehension over naive approaches. Empirical evaluation of such a subjective metric as comprehension is challenging due to the following reasons. 1) Viewers rarely share a common definition of coherence. They cannot be asked directly to give judgement on coherence. 2) Viewers differ in their per-

Manuscript received September 18, 2009; revised February 18, 2010; accepted March 01, 2010. Date of publication March 25, 2010; date of current version June 16, 2010. This work was supported by the National Science Foundation CAREER award 0092586.

A. Jhala is with the Jack Baskin School of Engineering, University of California, Santa Cruz, CA 95064 USA (e-mail: jhala@cs.ucsc.edu).

R. M. Young is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695 USA (e-mail: young@csc.ncsu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCIAIG.2010.2046486

[1] Darshak is a Sanskrit word meaning *viewer*.

1943-068X/$26.00 © 2010 IEEE

ceived coherence and the values reported by them cannot be directly mapped to a uniform scale. 3) Coherence is a property of both the cinematic discourse and the story plan itself.

In the remainder of this paper, we first present related work on narratology and intelligent camera control. Then, we discuss and motivate the central problem addressed by Darshak. We follow this discussion with the details of the planning algorithm and a detailed example of cinematic generation. Finally, we present an empirical evaluation of the system.

I. Related Work

The work described here employs a bipartite representation of narrative proposed by Chatman [11]. In Chatman's model, a narrative can be viewed as composed of two interrelated parts: the *story* and the *discourse*. The story describes the fictional world with all its content, characters, actions, events, and settings. The discourse contains medium-specific communicative content responsible for the *telling* of the narrative, for instance, a selection of a subset of events from the fabula, an ordering over these events for recounting, and linguistic communicative actions to tell the story. In this paper, our focus is on representation and reasoning about elements at the discourse level, specifically narrative discourse that is communicated visually through the use of cinematic conventions by moving a virtual camera in a 3-D environment rendered within a game engine.

One central problem in the automatic generation of cinematic narrative discourse is the selection of viewpoints in 3-D space that follows cinematic conventions and communicates the elements of the narrative unfolding in the 3-D environment. Previous work on intelligent camera control has focused mainly on the frame-by-frame graphical placement of the camera to satisfy given cinematic constraints [12]–[14]. Less attention has been paid to informing the placement of the camera over time based on the context of the story events [9], [15]. Evaluation of these camera systems has either been on runtime performance of algorithms [5], or user modeling based on specific interaction models in interactive scenarios [16]. The problem of automatically controlling the camera in 3-D environments has received significant attention from the research community [17]. Earliest approaches [18]–[20] focused on the mapping between the degrees of freedom (DOF) for input devices to 3-D camera movement.

Generalized approaches have modeled camera control as a constraint satisfaction or optimization problem. These approaches require the designer to define a set of required frame properties which are then modeled either as an objective function to be maximized by the solver or as a set of constraints that the camera configuration must satisfy. Optimization-based systems, like CAMPLAN [21], seek to find the best camera configuration for the given requirements. However, their computational cost is often prohibitive. On the other hand, constraint fulfilling systems [22] are much more efficient but may not return any result if there is no configuration respecting all the frame requirements.

Bares [23] addressed the issue by identifying conflicting constraints and producing multiple camera configurations corresponding to the minimum number of nonconflicting subsets. Bourne [24] extended Bares' solution by adding a *weight* property to each constraint to define a relaxation priority. A third set of generalized approaches [25]–[27] combines constraint satisfaction to select feasible 3-D volumes, optimizing search within selected volumes.

The low-level camera control system that is implemented as part of the Darshak system is based on Bares' ConstraintCam [12]. ConstraintCam is a real-time camera planner for complex 3-D virtual environments. It creates camera shots which satisfy user-specified goals in the form of geometric composition constraints. The system uses a modified constraint satisfaction problem (CSP) representation with camera position (x, y, z), view (xv, yv, zy), field of view (FOV) and up vector (dx, dy, dz) defined as the variables. The measure of satisfaction of a constraint is calculated as a fractional value between 0 and 1. The total satisfaction measure for a given solution is computed using the weighted sum of the satisfaction values of all the constraints.

With advances in narrative-oriented graphical world and their increasing use in training simulations and other pedagogically oriented applications, research initially considered the question of *how* to place the camera within a 3-D world. This question was addressed by the Virtual Cinematographer [28] system by formalizing *idioms*—stereotypical ways to film shots that were identified by cinematographers based on their filmmaking experiences. Other efforts used agent-based approaches, for instance, modeling the camera as an invisible creature [29] inside a 3-D world that responded to the emotion of the scene and its characters to select an appropriate camera shot.

More recent approaches to camera control (e.g., [8] and [9]), such as the one described in this paper, have started investigating the motivation behind *why* the camera is being used in a particular manner. Such approaches, mainly oriented towards narrative generation systems, consider the underlying context of the narrative to generate planned action sequences for the camera. They take into account the rhetorical coherence of these planned action sequences and the communicative intentions of the cinematographer during their construction. The challenges faced by these systems are similar to problems addressed by work on natural language discourse generation involving the selection of appropriate linguistic communicative actions in a textual discourse. In this community, theories of text discourse structures [30], [31] have been successfully used as the basis for the generation of coherent natural language discourse [32]–[34]. Integrating these theories with research on camera control has led to implementation of multimodal systems [35] that use the same underlying rhetorical structures to generate discourse in the form of natural language as well as directives for the camera.

II. Cinematic Narrative Discourse Generation

Cinematic discourse in the Darshak system is generated by a hierarchical partial-order causal link (POCL) planner whose abstract actions represent cinematic patterns and whose primitive actions correspond to individual camera shots. As shown in Fig. 1, the system takes as input an operator library, a story plan, and a set of communicative goals. The operator library contains a collection of action operators that represent camera placement

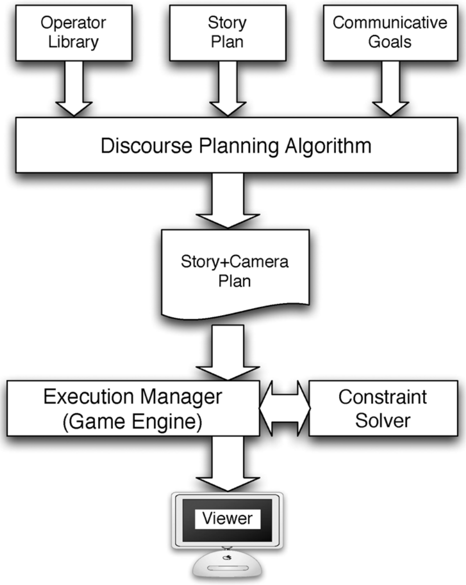

Fig. 1. Darshak architecture overview.

actions, transitions, abstract cinematic idioms, and narrative patterns. Camera placement and transition actions, represented as primitive operators, affect the focus of attention of the viewer and have preconditions that encode continuity rules in cinematography. Operators representing abstract cinematic idioms and narrative patterns affect the beliefs of the viewers and encode recipes for sequencing primitive or abstract operators. A description of the story to be filmed is input as a plan data structure that contains the description of the initial state of the story world, a set of goals achieved by the story characters' actions, and a totally ordered sequence of the story characters' actions and the causal relationships between them. The input story plan is added to the knowledge base for the discourse planner in a declarative form using first-order predicates that describe the elements of the data structure. The communicative goals given to the system describe a set of belief states to be established in the mind of the viewer.

The cinematic discourse planning algorithm performs both causal planning and temporal scheduling. To build a discourse plan, it selects camera operators from the operator library and adds them to the plan in order to satisfy specific communicative goals or preconditions of other communicative actions already in the plan. The algorithm binds variables in the camera operators, like start time and end time, relating the camera actions to corresponding actions in the story plan.

The output of the planning algorithm is a plan data structure containing a temporally ordered hierarchical structure of camera operators with all operator variables bound. The resulting plan is merged with the story plan and is sent to an execution manager running on the *Unreal Tournament* game engine.

A. Criteria for Effective Cinematic Narrative Discourse

Within the context of this work, *cinematic narrative discourse* is a recounting of events occurring in a 3-D graphical story world using a virtual camera. In our current work, we focus on specific elements considered by cinematographers to be central to a cinematic's effectiveness. In order to produce a system that generates effective cinematics, we define three properties of the narrative, viz., saliency, coherence, and temporal consistency, and have designed a cinematic discourse generation algorithm to produce cinematics that demonstrate these properties. While these properties have not been explicitly addressed in previous approaches to the automatic generation of story visualizations, we claim that they are central to the comprehension of cinematic narrative. Consequently, Darshak's design works to ensure that they are demonstrated in the cinematics it produces.

Selection of salient elements: Salient elements in a cinematic discourse are elements from the story (e.g., events, characters, objects, the relationships between them) that are relevant to inferences needed for comprehension. Choices made by a narrative generation system that determine which elements from the story are to be included in its telling directly affect salience and thus comprehension.

Plot coherence: Plot coherence can be described as the perception by the audience that the main events of a narrative are causally relevant to the story's outcome. [3]. In our work, plot coherence relates specifically to the perceived understanding of the causal relationships between events that a viewer constructs during narrative comprehension.

Temporal consistency: In this work, temporal consistency refers to consistency in the timing of the movement and positioning of the camera in relation to the events that are being filmed in the virtual world. Temporal consistency is closely linked to the established conventions of cinematography readily identified by viewers. Any inconsistencies in the timing or placement of a camera affect the communicative meaning of the shot as perceived by the audience.

B. Representation of Narrative Discourse in a Planning Formalism

As mentioned above, we adopt Chatman's [11] bipartite representation of narrative in our work. In this model, a narrative can be described as having a story—a set of causally related events, and a discourse—the telling of story elements through a specific medium. The frames, shots, and scenes described above operate at the discourse level but are dependent on the story for their content, drawn from the events, characters, settings, and objects appearing in them. Further, the organization of narrative discourse is shaped by information about relationships between different story world elements as well as the communicative intentions of the author/director.

The use of a hierarchical planning-based approach to cinematic discourse generation has several strengths. First, the capability of planning systems to reason about complex temporal and causal relationships between constituent actions in a plan allows them to compose cinematic plans that reflect both the complicated intentional relationships between cinematic segments and the causal and temporal relationships between camera actions and the underlying story actions they are responsible for filming. Second, the representation of abstraction over actions within hierarchical planners allows a system builder to readily represent both the hierarchical structure of a cinematic discourse as well as the patterns of cinematic action that are commonly used by cinematographers to film scenes.

While a planning approach offers these representational strengths, classical planning representations alone do not capture many of the essential elements of narrative discourse. For example, one assumption made in traditional planning is that actions execute instantaneously and assert their effects immediately. Another assumption often made is that any effect asserted by an action holds unless another action explicitly changes it. These assumptions are not always appropriate for the characterization of cinematic discourse. For instance, the effects of some camera actions hold only during the duration of the actions' execution, whether subsequent actions explicitly reverse those effects. To address these issues, Darshak's operator representation is annotated with temporal variables that are related to and constrained by events happening in the story world. We describe the elements of our representation in more detail below.

Typical POCL plan operators (e.g., those of UCPOP [36]) are defined using a collection of 1) constant symbols that identify objects in the domain, 2) variable symbols that range over the set of all objects in the domain, 3) constraints or filter conditions that describe invariant conditions in the world under which a given operator is applicable, 4) preconditions that identify changeable conditions in the world needed to hold immediately prior to an action in order for the action to successful execute, and 5) a set of effects that identify the exact ways that an action's successful execution changes the world. Darshak extends this representation in the following ways.

- *Variable symbols* are of two types, viz., object variables that range over the set of all constant symbols, and temporal variables that appear as elements in the specification of temporal constraints (described below).
- *Temporal constraints* are explicit specifications of the timing relationships between the times named by temporal variables (for instance, between execution times of camera and story actions). Relationships between conditions that describe temporal constraints are restricted to two types: precedence (indicated using the $<$ symbol) and equality (indicated using the $=$ symbol). This restriction enables us to represent temporal constraints efficiently via a graph with directed edges representing explicit relationships.
- *Temporally indexed conditions* are of the form

$$P(x_1, x_2, x_3, \ldots, x_n)@[t_s, t_e) \qquad (1)$$

where P is a predicate with arguments x_1 through x_n that holds for time t such that $t_s \leq t < t_e$.

1) Declarative Story Representation: The story provided to Darshak as input is given itself in the form of a POCL-style plan data structure and contains domain predicates, instantiated actions (plan steps), causal links between actions, ordering constraints on those actions, and goals of the story. This plan data structure is translated into a first-order representation of its elements and added to the camera planner's knowledge base as domain information for the camera planning problem. Table I shows the representation of a fragment of a story as input to the camera planning algorithm.

TABLE I
Story Plan Description Given as an Input to the Camera Planner

```
(character Royce) (character Marguerite)
(object phone)
(location room_royce) (location kitchen_marge)
(at Royce room_royce)
(at Marguerite kitchen_marge)
(mood Royce neutral (before s1)) (mood Marguerite neutral (before s1))
(conversation c1) (conv-type c1 phone) (conv-start c1 s1) (conv-end c1 s2)
(conv-steps c1 (s1 s2 s3))
    (step s1) (act-type s1 speak) (agent s1 Royce) (secondary s1 Marguerite)
    (effect s1 (spoken Royce ”What the . . . Hello”))
    (mood Royce neutral (during s1))
    (step s2) (act-type s2 speak) (agent s2 Marguerite) (secondary s2 Royce)
    (mood Marguerite neutral (during s2))
    (effect s2 (spoken Marguerite ”Do I hear lord’s name in vain”))
```

TABLE II
LookAt Operator and Corresponding Shot

Operator **Type :** LookAt **Parameters :** ?f, ?shot-type, ?dir **Preconditions :** none **Constraints :** (>Tend Tstart) **Effects :** (infocus ?f)@[Tstart, Tend)	

This story plan representation is used in the definitions of camera operators to describe their applicability conditions, constraining the selection of camera actions for use when filming different story elements. In the following sections, we describe our formalization of communicative operators used by Darshak across three hierarchical layers, viz., primitive shots, episodes/abstract film idioms, and dramatic patterns. A library of communicative operators composed from these three layers is an additional input to the camera planning algorithm (Fig. 1).

2) Primitive Shots: A *primitive shot* in Darshak defines the characteristics of an action to be carried out directly by the physical camera. For instance, the shot in Table II is a closeup of a character shown from the front. The operator that describes the shot is shown in Table II.

Fortunately for our representational needs, cinematographers typically make use of a relatively small number of primitive camera shots [37]. From a cinematographer's perspective, primitive shot specifications describe the composition of the scene with respect to the underlying geometric context. The main goal of a cinematographer when choosing a primitive shot is to compose the shot such that the viewer focuses on a certain aspect of the scene that is being framed. Through a sequence of coherent focus shifts, scenes are built that communicate the story. Represented as plan operators, primitive shot definitions capture certain coherence and shot-composition rules. For example, in order to avoid disorienting the viewer, one of the preconditions of a tracking shot—a shot where the camera moves relative to the movement of the view target—is that the actor or object that the operator is tracking should be in focus before movement starts. Adding a precondition (infocus ?f)@[Tstart) to a tracking camera shot operator ensures that a jump to tracking camera will not occur if it involves a focus shift from another object or from a different shot composition.

Composition and focus of primitive shots contributes to the overall dramatic presentation of the story. Van Sijll [39] lists

100 camera shot conventions that are used to convey various dramatic elements in movies. Each shot's dramatic value is described with respect to its screenplay and blocking. Darshak's representation of operators captures a subset of these dramatic effects of camera actions. Nine primitive operators and several variations of these operators are implemented in the system. For instance, three variants of the LookAt operator are: LookAt-Close, LookAt-Medium, and LookAt-LongShot that view an actor or object from progressively farther distances with appropriate cinematic framing for each distance.

3) Episodes/Abstract Scenes: Film idioms lend themselves to a hierarchical representation, with subparts that are themselves characterizations of idioms and a reduction that terminates in the types of primitive shots described above. While primitive shot types determine viewer's focus, abstract shot types represent the effect of focus and focus shifts on the mental states of the viewer. Individual shots like the ones described in Section II-B2 have a denotative meaning associated with them that focus the viewer's attention on elements in the frame. Abstract shot types also encode relationships *between* primitive operators and provide a mechanism for expressing established idioms as specific recipes or decompositions. Abstract plan operators, such as those used in a typical hierarchical task network (HTN)-style planning [40], are used in Darshak to capture such communicative phenomena by explicitly representing the different ways in which sequences of more primitive shots can convey discourse-level information about the story being filmed.

4) Dramatic Patterns in Narrative: As described earlier, there are established patterns of effective storytelling that have been documented by narrative theorists. These patterns implicitly manipulate focus of attention in effective ways. These narrative patterns are also operationalized as plan operators in Darshak. This representation allows the operator designer to specify constraints on salient elements according to their preferences.

There are two features of narrative patterns that facilitate their formalization into a computational model. First, the patterns are described using parameters and collections of properties of those parameters. For instance, in the pattern Deliverance (Table III), there are three participants, each with distinct roles (i.e., sinner, punisher). Participants are referred to in this operator schematically via parameters, as the definition refers not to specific individuals but to role fillers that can be mapped to characters across many stories. These parameters and their characteristics guide the story director to establish certain characteristics of each character he or she introduces. The restrictions on parameters also constrain choice of character actions that can be filmed in order to explicitly achieve the communicative goals of the dramatic pattern. Another important feature of narrative patterns is that there are clearly documented variations of each pattern and their communicative effect. So, while the pattern definition is general, variations in realization are also described.

TABLE III
OPERATOR DEFINITION OF *DELIVERANCE* PATTERN

Deliverance (?unfortunate, ?threatener, ?rescuer)
Description: An unfortunate individual/group under control
of a threatener is protected by an unexpected rescuer.

Constraints:
!Brave(?rescuer)
(storyaction ?threateningaction)
(effect ?threateningaction (Threatens(?threatener, ?unfortunate)))
(storyaction ?failedattemptaction)
(type-of ?failedattemptaction FailedEscape)
(storyaction ?rescueaction)
(effect ?rescueaction (Rescued(?unfortunate, ?threatener, ?rescuer)))

Preconditions:
Bel(V, Threatens(?threatener, ?unfortunate, ?threateningaction))
Bel(V, FailedEscape(?unfortunate, ?failedattemptaction))
Bel(V, ! Brave(?rescuer))

Effects:
Bel(V, Rescued(?unfortunate, ?threatener, ?rescuer, ?rescuedaction))

Steps:
Falling-Prey-To-Misfortune (?unfortunate)
  Preconditions: Bel(V, Fortunate(?unfortunate))
  Effects: Bel(V, Unfortunate(?unfortunate))
Show-Threatening-Action(?threatener, ?unfortunate, ?threateningaction)
Show-Failed-Escape(?threatener, ?unfortunate, ?failedattemptaction)

Using these two features, it is possible to describe narrative patterns as hierarchical plan operators. The parameters of the pattern are captured using the parameters of the operator. The role of each parameter within the context of the story is represented by constraints on the parameters of the operator. The variations of each pattern can be expressed in terms of collections of alternative decomposition operators, all decomposing the same abstract action but each with varying sets of applicability constraints.

The set of dramatic patterns that form the basis of Darshak's operator representation were identified by Polti based on exhaustive analysis of a large corpus of popular stories [41], ranging in genre from Greek mythology to late 19th century literature. These patterns were initially identified and individually described in much detail. Subsequent to their definition, however, there has not been an attempt to describe the relationships *between* these patterns. In our work, patterns identified by Polti are extended with the addition of relationships between patterns using constructs from plan operators (specifically, preconditions and ordering constraints on the patterns). These relationships are expressed as causal links, ordering links, and decomposition links within a recipe (that is, a specification of the means by which the abstract action's effects are to be achieved by a particular subplan). The main rationale for choosing Polti's patterns as the basis for our formalism is because these patterns lend themselves to direct formalization as plan operators with character roles represented in constrained parameterized scenes. The computational techniques described in this paper are independent of the underlying theory so long as the theory provides a basis for representing actors, actions, causality, temporality, and constraints on these elements of the story.

To better illustrate the formalization of dramatic patterns, consider the operator representation of the pattern *Deliverance* shown in Table III. Deliverance, as described by Polti, occurs when an unfortunate individual or group commanded by a threatener is rescued by an unexpected rescuer. This pattern is common in stories with a protagonist who rises above his or her capabilities to perform a heroic rescue. The parameters for deliverance are: the unfortunate, the threatener, and the rescuer. The communicative recipe for Deliverance involves making the user believe that 1) there is a threatener who is controlling the unfortunate, 2) the unfortunate does not have hope of escaping, and 3) an unexpected rescuer executes an action that leads to the threatener being overpowered and the unfortunate rescued.

In order to communicate deliverance effectively, a director first needs to satisfy certain preconditions regarding the beliefs of the viewer. One such precondition is that the viewer should believe that the protagonist cannot be expected to succeed in rescuing the unfortunate. This can be achieved using other patterns, for instance, ones that prompt the viewer when to expect the rescuer to fail by showing examples of his or her earlier failed attempts. The operator representation of Deliverance is shown in Table III.

C. DPOCL-T Algorithm for Generating Cinematic Discourse

In addition to their use in generating plans for physical activity (e.g., robot task planning), planning algorithms have been successfully used in the generation of effective textual discourse [34] as well as for story generation [43]. As described in the previous section, the representation of narrative discourse operators in Darshak encodes a rich formal representation of the causal structure of the plan. Each dependency between goals, preconditions, and effects is carefully delineated during the plan construction process. The system searches through the space of all possible plans during the construction process and thus can characterize the plan it produces relative to the broader context of other potential solutions to its planning problems.

To build Darshak, we extended our earlier work [42] on the DPOCL hierarchical planning algorithm to create decompositional POCL planning algorithm with temporal constraints (DPOCL-T). DPOCL-T forms the core planning algorithm used in Darshak.[2] The following section provides formal definitions of the constituent parts that make up a DPOCL-T planning problem.

1) Action Representation: DPOCL-T supports durative actions with temporal constraints on temporal variables. Actions are defined using a set of action schemata consisting of an action type specification, a set of free object variables, a set of temporal variables, a set of temporally indexed preconditions of the action, a set of temporally indexed effects, a set of binding constraints on the variables of the action, and a set of temporal constraints on the temporal variables of the action.

This action representation differs from previous approaches to discourse planning in its explicit representation of temporal variables and constraints on these variables. The set of temporal variables implicitly contains two distinguished symbols that denote the start and the end of the action instantiated from the schema in which they occur.

Actions that are directly executable by a camera are called *primitive*. Actions that represent abstractions of more primitive actions are called *abstract* actions. Abstract actions have zero or more *decomposition schemata*; each decomposition scheme for a given abstract action describes a distinct recipe or subplan for achieving the abstract action's communicative goals.

D. Domain, Problem, and Plan

A planning *problem* in DPOCL-T specifies a starting world state, a partial description of desired goal state, and a set of operators that are available for execution in the world. Since conditions are temporally indexed, the initial state of the *camera* problem not only specifies the state of the world at the beginning of execution but also indicates the *story* actions and events that occur in the future (that is, during the temporal extent of the camera plan). As a result, DPOCL-T is able to exploit this knowledge to generate camera plans that carefully coordinate their actions with those of the unfolding story. DPOCL-T plans are similar to those built by DPOCL except for the manner in which temporal ordering is specified. In DPOCL, the relative ordering of steps is expressed through explicit pairwise ordering links defining a partial order on step execution. In DPOCL-T, the ordering is implicitly expressed by the temporally constrained variables of the steps.

E. DPOCL-T Algorithm

Given a problem definition as described in the previous section, the planning algorithm generates a space of possible plans whose execution starting in the problem's initial state would satisfy the goals specified in the problem's goal state. The DPOCL-T algorithm generates plans using a refinement search through this space. The algorithm is provided in Fig. 2. At the top level, DPOCL-T creates a graph of partial plans. At each iteration, the system picks a node from the fringe of the graph and generates a set of child nodes from it based on the plan it represents. Plans at these child nodes represent single-step refinements of the plan at their parent node. Plan refinement involves either causal planning, where steps' preconditions are established by the addition of new preceding steps in the plan, or episodic decomposition, where subplans for abstract actions are added to the plan. The final step of each iteration involves checks to resolve any causal or temporal inconsistencies that have been added to the child plans during plan refinement. Iteration halts when a child plan is created that has no flaws (that is, that has no conflicts, no preconditions that are not causally established and no abstract steps without specified subplans).

1) Causal Planning: In conventional planning algorithms, an action is added to a plan being constructed just when one of the action's effects establishes an unestablished precondition of another step in the plan. To mark this causal relationship, a data structure called a causal link is added to the plan, linking the two steps and that condition that holds between their relative execution times.

In DPOCL-T, causal planning also requires enforcing temporal constraints on the intervals in which the relevant preconditions and effects hold. When Darshak establishes a causal link between two steps s_1 and s_2, additional constraints are also added to the plan. Suppose that s_1 (ending its execution at time t_1) establishes condition p needed by a precondition of s_2 at time t_2. When the causal link between s_1 and s_2 is added, Darshak also updates its constraint list to include the constraint $t_1 \leq \mathrm{t} < t_2$. Further, because p must now be guaranteed to hold

[2] Space limitations prevent us from providing a full discussion of the DPOCL planning algorithm. For more details, see [42].

DPOCL-T ($P_C = \langle S, B, \tau, L_C, L_D \rangle, \Lambda, \Delta$)

Here P_C is a partial plan. Initially the procedure is called with S containing placeholder steps representing the initial state and goal state and O containing temporal constraints on the initial and goal state time variables.

Termination: If P_C is inconsistent, fail. Otherwise if P_C is complete and has no flaws then return P_C

Plan Refinement: Non-deterministically do one of the following

1) Causal planning
   a) Goal Selection: Pick some open condition p@[t) from the set of communicative goals
   b) Operator Selection: Let S' be the step with an effect e that unifies with p. If an existing step S' asserts e then update the temporal constraints for t to be included in the protection interval for the effect e. If no existing step asserts e then add a new step S' to the plan and update the temporal constraint list with the variables introduced by step S'
2) Episode Decomposition
   a) Action Selection: Non-deterministically select an unexpanded abstract action from P_C
   b) Decomposition Selection: Select a decomposition for the chosen abstract action and add to P_C the steps and temporal and object constraints specified by the operator as the subplan for the chosen action.

Conflict Resolution:
For each conflict in P_C created by the causal or decompositional planning above, resolve the conflict by nondeterministically chosing one of the following procedures:

1) Promotion: Move S1 before S2 (Add constraints on the start and end time points of S1 and S2)
2) Demotion: Move S2 before S1 (Add constraints on the start and end time points of S1 and S2)
3) Variable Separation: Add variable binding constraints to prevent the relevant conditions from unifying

Recursive invocation: Call **DPOCL-T** with the new value of P_C.

Fig. 2. Sketch of the camera planning algorithm.

between t_1 and t_2, Darshak checks at each subsequent iteration that this constraint holds. In this case, we adopt terminology used by [44] and call the interval $[t_1, t)$ a *protection interval* on the condition p_i.

2) Decompositional Planning: Decomposition schemata in DPOCL-T are similar to their counterparts in DPOCL, both in their content and in their manner of use. The sole point of difference is in the schemata's representation of temporal constraints—rather than include a set of explicit pairwise orderings between steps within a decomposition schema, DPOCL-T specifies partial orderings via constraints on the time variables determining the decomposition's constituent steps' execution.

3) Management of Temporal Constraints: As new steps are added to a plan being built, the effects of these steps may threaten the execution of other steps in the plan. That is, their effects might undo conditions established earlier in the plan and needed by the preconditions of some steps later in the plan. In DPOCL-T, a global list of temporal constraints is maintained as a directed graph of all the time variables involved in the plan's current steps. Whenever the planner adds a step, all the steps' time variables are added to the directed graph and the graph is completed with the steps' new constraints representing the edges of the graph. Another step's precondition is threatened by a newly added step just when the effect of the new step changes the needed condition within its protection interval. Such threats are detected by identifying a cycle in the temporal constraints graph.

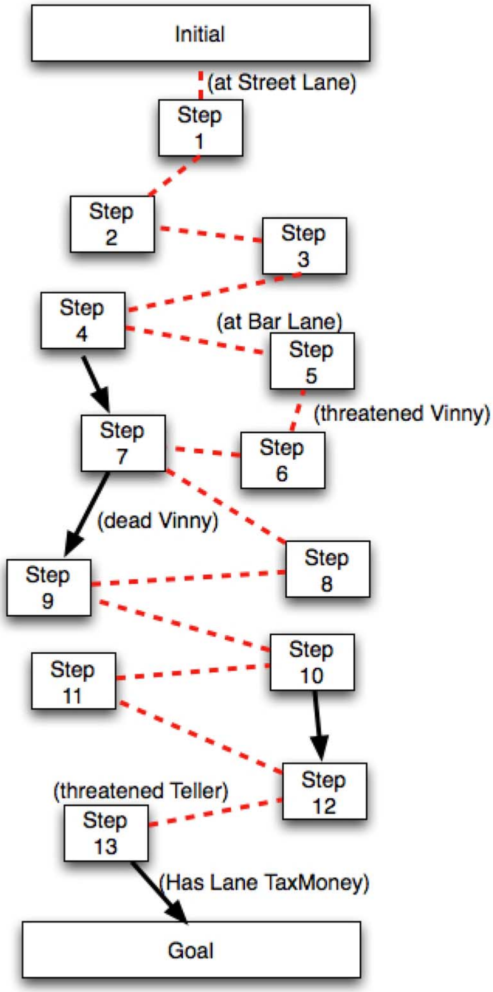

Fig. 3. Simplified pictorial representation of the example story plan input to the planner that is shown in Table IV. Boxes represent story actions, red dotted lines indicate ordering links, and solid black arrows denote causal links. Several causal links are labeled with conditions responsible for relationship between actions. The discourse plan for this story is shown in Table IV.

The Darshak algorithm produces plans offline from given stories and we are concerned with the quality of generated plans rather than fast execution of the algorithm. The exponential explosion of the plan space is mitigated by careful design of operators with variable binding and temporal constraints. For complex domains, the hierarchical representation of operators makes this approach scalable. There is a significant authorial burden in writing domain specifications that can be eased through storyboarding interfaces.

III. An Example of Cinematic Discourse Generation in Darshak

The DPOCL-T algorithm is illustrated by the following story about a thief, Lazarus Lane, who goes to a town in Lincoln County, NV, to steal the tax money that is stored in the local bank. In the story, Lane successfully steals the money after winning Sheriff Bob's favor and being appointed as the deputy. The entire input story is shown in Table IV. The initial state for the

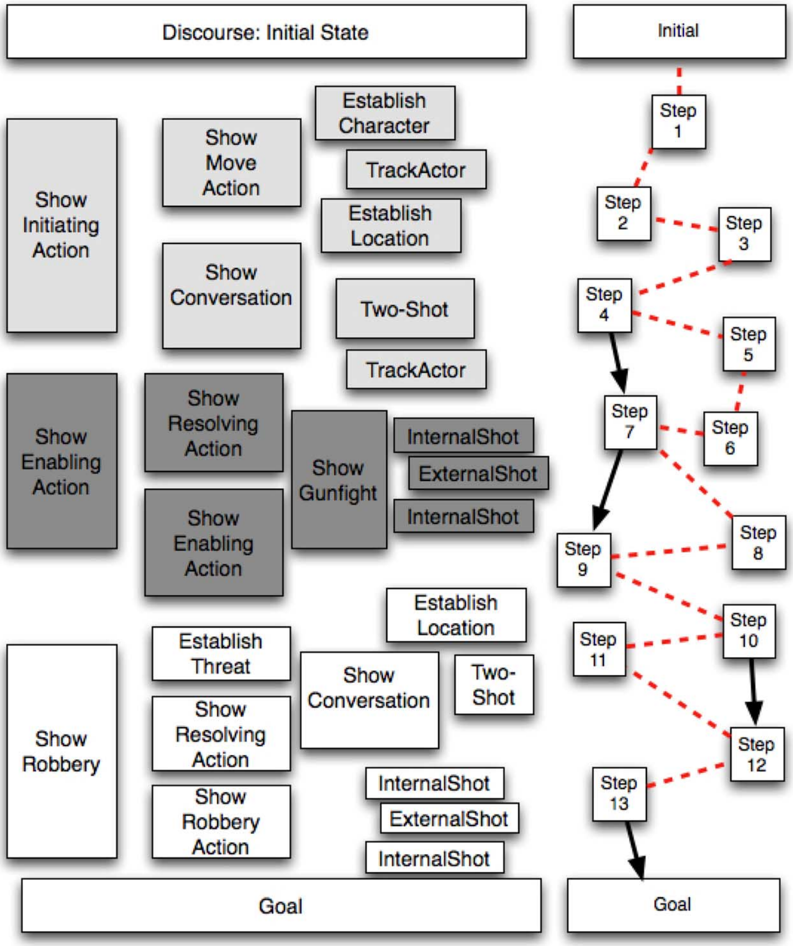

Fig. 4. Fully realized discourse plan for the example story. The story plan steps are shown on the right. The discourse plan actions are shown hierarchically from left to right. Right most actions are primitive actions and actions to the left are abstract actions. Primitive camera actions film the story world actions that they are adjacent to in the figure. Dotted red lines show the temporal sequence of actions from top to bottom and black arrows depict causal links between story world actions.

TABLE IV
EXAMPLE STORY

Step 1 Lane goes to the Bar
Step 2 Lane asks the bartender for a drink
Step 3 Lane overhears that Vinny the outlaw has murdered the town's deputy sheriff
Step 4 Lane overhears that Sheriff Bob has cancelled the plans for going out of town as the town is without a deputy
Step 5 Lane goes to see Vinny
Step 6 Vinny threatens Lane
Step 7 Lane Shoots Vinny
Step 8 Lane goes to Sheriff
Step 9 Sheriff Bob appoints Lane as deputy sheriff
Step 10 Sheriff Bob leaves town
Step 11 Lane goes to the bank
Step 12 Lane Threatens the Teller
Step 13 Teller gives Lane the tax money from the bank vault

discourse planning problem contains sentences about the story actions. The goal state contains the lone goal for the discourse planner, (BEL V (HAS TAXMONEY LANE)). It is assumed that the viewer has no prior beliefs about the story world. The discourse generated by the planner communicates to the viewer how Lane successfully steals the tax money from Lincoln county.

Initially, the goal of the planning problem is established by instantiating a new discourse action in the plan: SHOW-ROBBERY(LANE, TAXMONEY, BANK). This action is chosen because it has an effect (BEL V (HAS LANE TAXMONEY)). From the planner's perspective, the SHOW-ROBBERY action is causally motivated by the open condition of the goal state. The abstract action selected by the planner represents one of the narrative patterns that can be used to achieve the goal of telling the viewer the story of a character obtaining an object through a sequence of actions in the story world.[3] Given this choice, the planner instantiates the action and adds it to the discourse plan, and updates the open conditions list with the preconditions of the SHOW-ROBBERY action.

The abstract action is then decomposed in the next planning step into constituent actions. In this example, the SHOW-ROBBERY action is expanded to three sub-actions: SHOW-ROBBERY-ACTION, SHOW-THREAT, and SHOW-RESOLUTION. The discourse actions are bound to story actions through the instantiation of constraints on the operators. The SHOW-ROBBERY-ACTION, in this case, has constraints that bind the story step that this action films. The step in the story

[3] At this point, the planner could have chosen any of the other patterns with the effect (BEL V (HAS ?CHARACTER ?OBJECT)) that would have also satisfied the story world's constraints.

plan that indicates successful robbery is Step 13, where the teller gives Lane the tax money from the vault. This action has the effect (Has Lane taxMoney). The corresponding constraint on the Show-Robbery-Action operator is (effect ?s (Has ?char ?obj)) which binds ?s to Step13, ?char to Lane, and ?obj to taxMoney. In this way, constraints on the discourse operators are checked by queries into the knowledge base describing story plan; as a result, correct bindings for the story world steps are added to the plan structure. Once the step is correctly bound, the start and end temporal variables for the Show-Robbery-Action are bound to (start Step13) and (end Step13), respectively, and the temporal constraint graph is updated with these new variables. The planning process continues with expansion of abstract actions and addition of new actions to satisfy open conditions.[4] Fig. 4 illustrates a complete camera plan.

During the addition of temporal constraints to the discourse plan, the planner maintains a simple temporal graph [45] of all the steps' time variables and checks for consistency of the graph after the addition of each action. Each temporal variable has a protection interval during which it satisfies all the constraints imposed on it by the camera operators. As actions are added to the plan, the temporal consistency checking algorithm constantly updates this protection interval and checks for invalid intervals to prune illegal temporal orderings for the actions.

Once a complete discourse plan is created, Darshak's output is sent to an execution component that runs the plans within a 3-D game engine. This execution management is carried out within the Zocalo framework [46]. The system is built upon the Zocalo service-oriented architecture developed as a research testbed for interactive storytelling. Zocalo consists of a server running the hierarchical POCL planner—Darshak—through a web-service called Fletcher [46]. The execution of planned stories and camera sequences occurs on a game engine (*Unreal Tournament 2004*) through a lightweight scheduler that communicates with the web service. Sample output of Darshak on the game engine is shown in Fig. 5.

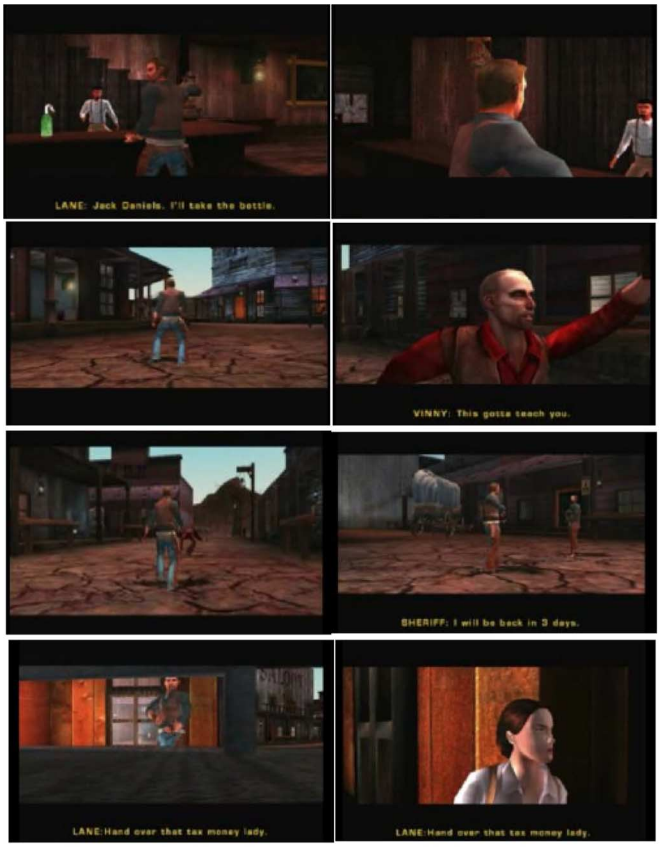
LANE: Jack Daniels. I'll take the bottle.
VINNY: This gotta teach you.
SHERIFF: I will be back in 3 days.
LANE:Hand over that tax money lady.
LANE:Hand over that tax money lady.

Fig. 5. Snapshots of Darshak's output for the example story rendered on the game engine.

IV. Empirical Evaluation

To evaluate the effects of different visualization strategies, three visualizations of the same story were prepared: one with a fixed camera position within the setting, one with an over-the-shoulder camera following the protagonist, and one with a camera plan automatically generated by Darshak. The purpose for running these experiments was twofold: first, to investigate whether visualization strategies do indeed affect comprehension; and second, to evaluate the quality of visualization generated by Darshak using a representation of camera shots as communicative actions. The objective of the experiments was to determine whether visualizations generated by Darshak are coherent (as measured by viewers' perceptions of the attributes of the underlying stories).

Our experimental approach was to first map the stories being shown to subjects into a representation previously developed and validated as a cognitive model of narrative. We then probed subjects' understanding of the narrative that was presented in a cinematic to determine how closely their understanding aligned with the cognitive model's predictions about a viewer's mental model. As the underlying story elements in this system were defined as plan data structures themselves, the experimental design was based on previous work [7] relating these data structures to the mental models that users form during comprehension. To do this, a mapping is defined from plan data structures onto a subset of the conceptual graph structures adopted from the QUEST model, developed by Graesser *et al.* [6]. In the QUEST model [6], stories are represented as conceptual graph structures containing concept nodes and connective arcs. Together with a specific arc search procedure, QUEST was originally used to provide a cognitive model of question answering in the context of stories (supported question types include why, how, when, enablement, and consequence questions). In our work, we make use only of the graph structures, referred to here as QUEST knowledge structures (QKSs). They describe the reader's conception of narrative events and their relationships. The composition of nodes and arcs in a QKS structure reflects the purpose of the elements they signify in the narrative. For instance, if nodes A and B are two events in a story such that A causes or enables B, then A and B are represented by nodes in the QKS graph and are connected by a Consequence type of arc.

Techniques used by Graesser *et al.* to validate the QUEST model were based on goodness-of-answer (GOA) ratings for question–answer pairs about a story shown to readers. Subjects

[4] As part of the planning process, the planner creates an entire *space* of possible plans that might solve its planning problem. The example here traces a single path through the construction process leading to the correct solution.

were provided with a question about the story and an answer to the question, then asked to rate how appropriate they thought the provided answer was given the story that they had read. GOA ratings obtained from their subjects were compared to ratings predicted by the arc search procedures from QUEST model. Results from their experiments clearly showed a correlation between the model and the cognitive model developed by readers to characterize stories.

The algorithm for converting a POCL plan data structure to the corresponding QKS structure is adopted from Christian and Young [7]. In this experiment, first the story, represented as a plan data structure, is converted to a corresponding QKS structure. Predictor variables proposed in the QUEST model are used to calculate predictions for the GOA ratings—the measure of goodness of answer for a question–answer pair related to the events in the story. These GOA ratings are compared against data collected from participants who watch a video of the story filmed using different visualization strategies. The three predictors that are correlated to the GOA ratings are arc search, constraint satisfaction, and structural distance. Correspondence between GOA ratings indicated by these predictor variables and by human subjects would be taken as an indication that the cinematic used to tell the story had been effective at communicating its underlying narrative structure.

A. Method

1) Design: To create story visualizations, we used two stories (S1 and S2) and three visualization strategies for each story (V1-fixed camera, V2-over-the-shoulder camera angle, and V3-Darshak driven camera), creating six treatments. Treatments were identified by labels with story label as prefix followed by the label of the visualization. For instance, S2V1 treatment would refer to a visualization of the second story (S2) with fixed camera angle strategy (V1). Participants were randomly assigned to one of six groups (G1 to G6). Thirty participants, primarily undergraduate and graduate students from the Computer Science Department, North Carolina State University (Raleigh, NC), participated in the experiment. Each participant was first shown a video and then asked to rate question–answer pairs of three forms: how, why, and what enabled. The process was repeated for each subject with a second video.

A Youden squares design was used to distribute subject groups among our six treatments. This design was chosen in order to account for the inherent coherence in the fabula and to account for the effects of watching several videos in order. Assuming a continuous response variable, the experimental design, known as a Youden square, combines Latin squares with balanced, incomplete block designs (BIBD). The Latin square design is used to block on two sources of variation in complete blocks. Youden squares are used to block on two sources of variation—in this case, story and group—but cannot set up the complete blocks for Latin squares designs. Each row (story) is a complete block for the visualizations, and the columns (groups) form a BIBD. Since both group and visualization appear only once for each story, tests involving the effects of visualization are orthogonal for those testing the effects of the story type; the Youden square design isolates the effect of the visual perspective from the story effect.

TABLE V
2×3 YOUDEN SQUARES DESIGN FOR THE EXPERIMENT. G1 THROUGH G6 REPRESENT SIX GROUPS OF PARTICIPANTS WITH FIVE MEMBERS IN EACH GROUP. THEY ARE ARRANGED SO THAT EACH STORY AND VISUALIZATION PAIR HAS A COMMON GROUP FOR OTHER VISUALIZATIONS

Viz	**Master Shot**	**Over The Shoulder**	**Darshak**
S1	G1,G4	G2,G5	G3,G6
S2	G5,G3	G6,G1	G4,G2

Each story used in the experiment had 70 QKS nodes. Of the 70 QKS nodes, ten questions were generated from randomly selected QKS elements and converted to one of the three question types supported by QUEST: how, why, and what enabled. For each of the ten questions, approximately 15 answer nodes were selected from nodes that were within a structural distance of three nodes in the QKS graph generated from the story data structure. These numbers were chosen to have similar magnitude to Christian and Young's previous experiments, for better comparison.

2) Procedure: Each participant went through three stages during the experiment. The entire experiment was carried out in a single session for each participant. Total time for a single participant was between 30 and 45 min. Initially, each participant was briefed on the experimental procedure and was asked to sign the consent form. They were then asked to read the instructions for participating in the study. After briefing, they watched a video of one story with a particular visualization according to the group assignment (Table V). For each video, users provided GOA ratings for the question–answer pairs related to the story in the video. Participants were asked to rate the pairs along a four point scale: good, somewhat good, somewhat bad, bad. This procedure is consistent with earlier experiments [6], [7]. Next, they watched a second video with a different story and visualization followed by a questionnaire about the second story. The videos were shown in different orders to common groups in order to account for discrepancies arising from the order in which participants were shown the two videos.

B. Results and Discussion

The mean overall GOA ratings recorded for the two stories are shown in Table VI along with the standard deviations. These distributions of GOA scores do not present any problem for multiple regression analyses as the means do not show ceiling or floor effects. The standard deviations are high enough to rule out the potential problem of there being a restricted range of ratings. The GOA numbers shown in Table VI indicate on preliminary observation that the GOA ratings for V1 (master shot) and V3 (Darhsak) are significantly correlated with V2 (over-the-shoulder shots). The standard deviations for V3 are lower than the other treatments in both stories. This indicates that participants converge better on rating questions in Darshak-generated visualizations.

An interesting observation for V2 is that in story 2 the mean GOA ratings are significantly lower than the other two treatments with a significantly high standard deviation. These numbers support the intuition that participants form their own interpretation of events in the story while looking at shots that are over-the-shoulder leading to the wide disparity in ratings in going from story 1 to story 2. While mean ratings provide an

TABLE VI
MEAN GOA RATINGS AND STANDARD DEVIATIONS FROM THE EXPERIMENT

GOA(stddev)	V1	V2	V3
S1	1.69 (±0.91)	1.74 (±0.82)	1.70 (±0.79)
S2	1.76 (±0.59)	1.51 (±0.67)	1.78 (±0.59)

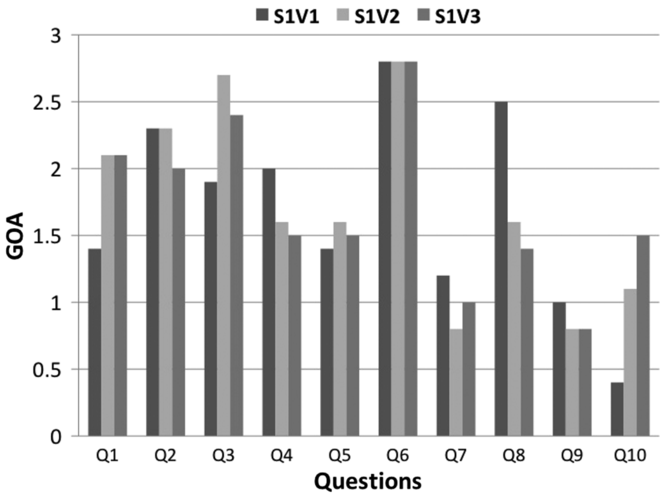

Fig. 6. GOA ratings for story 1 across the three visualization strategies (S1 and S2 are stories, V1-master shot,V2-over-the-shoulder shot, and V3-Darshak are visualization strategies).

overall idea of the participant's responses, GOA ratings for individual questions across different visualizations provide more insight into the differences across visualizations. Fig. 6 summarizes mean GOA ratings for individual questions related to story 1 for the three visualization treatments. Question numbers 1, 8, and 10 are particularly interesting as there is a big variation in the GOA ratings for the master shot visualization and the other two treatments, which have quite similar ratings. The question–answer pairs in discussion here are presented below.

1. Question: Why did Lane challenge Vinny?
   Answer: Because he wanted to kill Vinny.
8. Question: Why did Lane challenge Vinny?
   Answer: Because Lane wanted to steal tax money.
10. Question: Why did Lane meet Sheriff Bob?
    Answer: Becaue Lane needed a job.

In Q1 and Q10, the ratings for V1 are significantly lower. This could be explained by examining the relationships between the question–answer nodes. In all three cases, the question–answer nodes are two or more arcs away in distance along the causal chain of events. In case of the arc-search and structural distance predictors from QUEST these are good answers as they do lie on a causal chain of events leading to the question. The necessity and sufficiency constraints in the constraint satisfaction predictor reduce the strength of the answer. In Q1, for example, it is not necessary for Lane to challenge Vinny. He could just shoot him right away. In this case, viewers who were familiar with the gunfight setting chose to label the challenge as being an important step in killing Vinny as, for them, it forms an integral part of the gunfight sequence. In the master-shot visualization, the gunfight sequence was not even recognized as a gunfight by most participants (information obtained from postexperiment interview). This analysis indicates that additional consideration of the "why" type of questions on other nodes is needed to determine the effects of visualization strategies on GOA ratings related to perceived causal connections between events in the story.

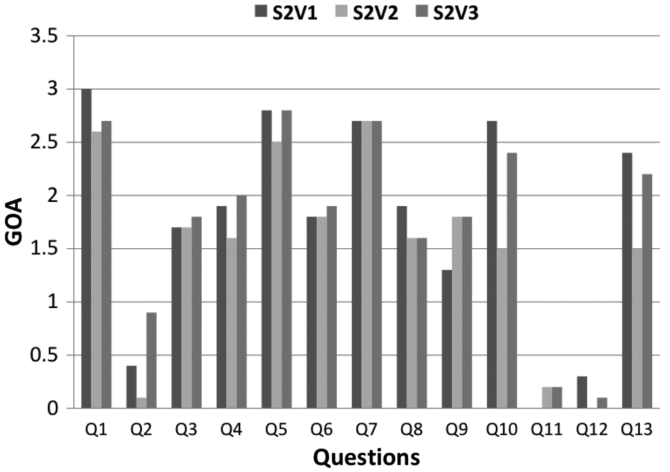

Fig. 7. GOA ratings for story 2 across the three visualization strategies (S1 and S2 are stories, V1-master shot,V2-over-the-shoulder shot, and V3-Darshak are visualization strategies).

Fig. 7 shows the average ratings for each question for the second story. The interesting responses are the ones that have a significant variation in mean ratings across different visualizations. In this story, unlike the data for story 1, the differences between ratings were relatively smaller. The interesting observations, however, were the ones where one of the treatments rated the answer as a "bad" answer (rating < 1.5) and the other treatments rated the answer as a "good" answer (rating > 1.5).

Postexperiment interviews were carried out after participants completed the rating forms for both stories. During these interviews, the participants were asked to give subjective answers to questions about the quality of the story and videos. The main purpose of these questions was to get additional information about metrics that were not considered in previous approaches but may play a role in analysis of GOA ratings. The data collected from this survey provide insight into a possible extension of the cognitive model of story understanding that take into account features of discourse that are not currently represented. Based on the subjective data, a majority of the subjects preferred system-generated videos to the other two videos. Most users reported that visualization did affect their engagement in the story. System-generated visualizations were rated as being more engaging. Users preferred camera movements over static camera shots as they perceived the scene to be more dynamic and interesting. While these qualitative responses are hard to measure statistically, they do point to uniformity among experiment participants regarding their preference for system-generated visualizations over other visualization strategies.

There are several caveats to the evaluation strategy based on cognitive models of story comprehension. The results of the study only address the cognitive aspect of story perception and do not directly measure specific discourse effects and aesthetic quality of produced output. Further work is needed within a richer story domain with opportunity to exploit subtleties of visual communication to fully evaluate the potential of Darshak's ability of generating aesthetic sequences. The current focus for Darshak, however, is on generating coherent sequences. A sound evaluation of coherence presented in this paper establishes Darshak's success for this focus.

V. Conclusion

The work described in this paper contributes to several different areas of research. First, Darshak's representation of shots and shot sequences moves beyond previous methods that focused on geometric constraints. Darshak's representation of film idioms provides higher level reasoning and extends constraint-based techniques by defining a relationship between low-level geometric constraints and abstract cinematic idioms. Darshak builds on prior approaches to modular constraint-based camera control by adding a richer representation of modules through their association with planning operators and uses that plan-based representation to reason about the effects of cinematic actions on the cognitive model of a viewer.

Second, Darshak's focus on camera control as narrative discourse demonstrates that visual communicative actions can be effectively represented through communicative plan operators and utilized for deliberate planning for cinematic storytelling. Darshak extends previous work on discourse planning by introducing cinematic idioms and camera shots as communicative operators. Further, Darshak builds on discourse planning approaches by introducing explicit representation of and reasoning about the temporal relationships between discourse actions and story actions—a feature critical for cinematic discourse generation.

Finally, we have introduced a novel evaluation method that is useful for determining the efficacy of approaches to automatic storytelling in a cinematic medium, especially relative to the cognitive models of story created by users during story comprehension. Prior work on evaluating cognitive models of narrative comprehension focused solely on the textual medium. Some work on evaluating computational models of narrative did focus on cinematics but did not take into account the range of visual communicative strategies produced in Darshak's experiments. Darshak's evaluation addresses both issues by using the Youden squares experimental design to explicitly measure communicative effects of various visualization strategies on the same underlying story. This study shows that GOA ratings obtained from participants viewing different visualizations of the same stories support our hypothesis that different visualization strategies indeed affect comprehension. Further, significant correlation between GOA ratings predicted by the QUEST predictors and two of the three visualization strategies provides support for the conclusion that stories communicated through automated discourse generated by Darshak are readily comprehensible to viewers.

References

[1] R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals and Understanding: An Inquiry Into Human Knowledge Structures*. Hillsdale, NJ: L. Erlbaum, 1977.

[2] J. R. Meehan, "Tale-spin, an interactive program that writes stories," in *Proc. Int. Joint Conf. Artif. Intell.*, 1977, pp. 91–98.

[3] M. O. Riedl and R. M. Young, "An intent-driven planner for multi-agent story generation," in *Proc. 3rd Int. Joint Conf. Autonom. Agents Multiagent Syst.*, New York, 2004, pp. 186–193.

[4] C. B. Callaway, "Narrative prose generation," *Artif. Intell.*, vol. 139, pp. 213–252, 2002.

[5] M. Christie, R. Machap, J. Normand, P. Olivier, and J. Pickering, "Virtual camera planning: A survey," in *Smart Graphics*, 2005, pp. 40–52.

[6] A. C. Graesser, K. L. Lang, and R. M. Roberts, "Question answering in the context of stories," *J. Exp. Psych., General*, vol. 120, no. 3, pp. 255–277, 1991.

[7] D. B. Christian and R. M. Young, "Comparing cognitive and computational models of narrative structure," in *Proc. 19th Nat. Conf. Artif. Intell.*, 2003, pp. 385–390.

[8] A. Jhala and R. M. Young, "Representational requirements for a plan-based approach to virtual cinematography," in *Proc. 2nd Conf. Artif. Intell. Interactive Digit. Entertainment*, Marina Del Rey, CA, 2006, pp. 36–41.

[9] A. Jhala and R. M. Young, "A discourse planning approach to cinematic camera control for narratives in virtual environments," in *Proc. 20th Nat. Conf. Artif. Intell.*, Pittsburgh, PA, 2005, pp. 307–312.

[10] G. Prince, *A Dictionary of Narratology*. Lincoln, NE: Univ. Nebraska Press, 1987.

[11] S. Chatman, *Story and Discourse: Narrative Structure in Fiction and Film*. Ithaca, New York: Cornell Univ. Press, 1980.

[12] W. H. Bares, S. McDermott, C. Boudreaux, and S. Thainimit, "Virtual 3D camera composition from frame constraints," in *ACM Multimedia*. New York: ACM Press, 2000, pp. 177–186.

[13] H. Nicholas, H. Ralf, and S. Thomas, "A camera engine for computer games: Managing the trade-off between constraint satisfaction and frame coherence," in *Proc. Eurographics*, 2001, vol. 20, pp. 174–183.

[14] D. B. Christianson, S. E. Anderson, L. W. He, D. Salesin, D. S. Weld, and M. F. Cohen, "Declarative camera control for automatic cinematography," in *Proc. AAAI/IAAI Nat. Conf. Artif. Intell.*, 1996, vol. 1, pp. 148–155.

[15] D. Amerson, S. Kime, and R. M. Young, "Real-time cinematic camera control for interactive narratives," in *Proc. ACM SIGCHI Int. Conf. Adv. Comput. Entertain. Technol.*, 2005, p. 369.

[16] W. Bares and J. Lester, "Cinematographic user models for automated realtime camera control in dynamic 3D environments," in *Proc. 6th Int. Conf. User Model.*, 1997, pp. 215–226.

[17] M. Christie and P. Olivier, "Camera control in computer graphics," *Eurographics*, vol. 25, no. 3, pp. 89–113, 2006.

[18] C. Ware and S. Osborne, "Exploration and virtual camera control in virtual three dimensional environments," *SIGGRAPH*, vol. 24, no. 2, pp. 175–183, 1990.

[19] J. Blinn, "Where am I? What am I looking at?," *IEEE Comput. Graph. Appl.*, vol. 8, no. 4, pp. 76–81, Jul. 1988.

[20] M. Gleicher and A. Witkin, "Through-the-lens camera control," in *Proc. 19th Annu. Conf. Comput. Graph. Interactive Tech.*, 1992, pp. 331–340.

[21] N. Halper and P. Olivier, "Camplan: A camera planning agent," in *Proc. AAAI Spring Symp. Smart Graph.*, 2000, pp. 92–100.

[22] F. Jardillier and E. Languènou, "Screen-space constraints for camera movements: The virtual cameraman," *Comput. Graph. Forum*, vol. 17, no. 3, pp. 175–186, 1998.

[23] W. H. Bares and J. C. Lester, "Intelligent multi-shot visualization interfaces for dynamic 3D worlds," in *Proc. 4th Int. Conf. Intell. User Interfaces*, 1999, pp. 119–126.

[24] O. Bourne and A. Sattar, "Applying constraint weighting to autonomous camera control," in *Proc. 1st Artif. Intell. Interactive Digit. Entertainment Conf.*, 2005, pp. 3–8.

[25] J. Pickering, "Intelligent camera planning for computer graphics," Ph.D. dissertation, Dept. Comput., Univ. York, York, U.K., 2002.

[26] M. Christie and J.-M. Normand, "A semantic space partitioning approach to virtual camera composition," *Comput. Graph. Forum*, vol. 24, no. 3, pp. 247–256, 2005.

[27] P. Burelli, L. Di Gaspero, A. Ermetici, and R. Ranon, "Virtual camera composition with particle swarm optimization," in *Smart Graphics*. New York: Springer-Verlag, 2008, pp. 130–141.

[28] H. Li-wei, C. Michael, and S. David, "The virtual cinematographer: A paradigm for real-time camera control and directing," in *Proc. SIGGRAPH Comput. Graphics Conf.*, 1996, pp. 217–224.

[29] B. Tomlinson, B. Blumberg, and D. Nain, "Expressive autonomous cinematography for interactive virtual environments," in *Proc. 4th Int. Conf. Autonom. Agents*, C. Sierra, M. Gini, and J. S. Rosenschein, Eds., Barcelona, Catalonia, Spain, 2000, pp. 317–324.

[30] B. J. Grosz and C. L. Sidner, “Attention, intentions, and the structure of discourse,” *Comp. Linguistics*, vol. 12, no. 3, pp. 175–204, 1986.

[31] W. Mann and S. Thompson, “Rhetorical structure theory: A theory of text organization,” USC/ISI Tech. Rep., 1987.

[32] R. M. Young, J. D. Moore, and M. E. Pollack, “Towards a principled representation of discourse plans,” in *Proc. Int. Conf. AI Planning Syst.*, Atlanta, GA, 1994, pp. 188–193.

[33] M. Maybury, “Communicative acts for explanation generation,” *Int. J. Man-Mach. Studies*, vol. 37, pp. 135–172, 1992.

[34] J. D. Moore and C. Paris, “Planning text for advisory dialogues: Capturing intentional and rhetorical information,” *Comput. Linguistics*, vol. 19, no. 4, pp. 651–694, 1994.

[35] C. B. Callaway, E. Not, A. Novello, C. Rocchi, O. Stock, and M. Zancanaro, “Automatic cinematography and multilingual NLG for generating video documentaries,” *Artif. Intell.*, vol. 165, no. 1, pp. 57–89, 2005.

[36] J. S. Penberthy and D. S. Weld, “UCPOP: A sound and complete partial-order planner for ADL,” in *Proc. Knowl. Represent. Reason.*, 1992, pp. 103–114.

[37] D. Arijon, *Grammar of the Film Language*. Los Angeles, CA: Silman-James Press, 1976.

[38] M. Joseph, *The Five C’s of Cinematography*. Hollywood, CA: Cine/Grafic, 1970.

[39] J. V. Sijll, *Cinematic Storytelling*. Studio City, CA: Michael Wiese Productions, 2005.

[40] E. D. Sacerdoti, “The nonlinear nature of plans,” in *Proc. Int. Joint Conf. Artif. Intell.*, 1975, pp. 206–214.

[41] G. Polti, *Thirty-Six Dramatic Situations*. Boston, MA: The Writer Inc., 1921, 1977.

[42] R. M. Young, M. E. Pollack, and J. D. Moore, “Decomposition and causality in partial-order planning,” in *Proc. AIPS*, Chicago, IL, 1994, pp. 188–193.

[43] M. Riedl and R. M. Young, “Character focused narrative planning,” in *Proc. Int. Conf. Virtual Storytelling*, Toulouse, France, 2003, pp. 47–56.

[44] D. Nau, M. Ghallab, and P. Traverso, *Automated Planning: Theory and Practice*. San Francisco, CA: Morgan-Kauffman, 2004.

[45] N. Yorke-Smith, K. B. Venable, and F. Rossi, “Temporal reasoning with preferences and uncertainty,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, pp. 1385–1386.

[46] T. Vernieri, “A web services approach to generating and using plans in configurable execution environments,” M.S. thesis, Dept. Comput. Sci., North Carolina State Univ., Raleigh, NC, 2006.

Arnav Jhala (M’99) received the B.Eng. degree in computer engineering from Gujarat University, Ahmedabad, India, in 2001 and the M.S. degree in computer science and the Ph.D. degree in computer science from the North Carolina State University, Raleigh, in 2004 and 2009, respectively.

He is an Assistant Professor of Computer Science at the Jack Baskin School of Engineering, University of California, Santa Cruz. His research interests lie at the intersection of artificial intelligence and digital media, particularly in the areas of computer games, cinematic communication, and narrative discourse. He has previously held research positions at the IT University of Copenhagen, Institute for Creative Technologies at University of Southern California where he was part of the Leaders project developed in association with Paramount Pictures. He has also worked on the America’s Army: Adaptive Thinking and Leadership game at Virtual Heroes, Inc., a leading serious games developer and at the Indian Space Research Organization (ISRO).

R. Michael Young (M’98–SM’07) received the B.S. degree in computer science from the California State University at Sacramento, Sacramento, in 1984, the M.S. degree in computer science with a concentration in symbolic systems from Stanford University, Stanford, CA, in 1988, and the Ph.D. degree in intelligent systems from the University of Pittsburgh, Pittsburgh, PA, in 1998.

He worked as a Postdoctoral Fellow at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. He is currently an Associate Professor of Computer Science and Co-Director of the Digital Games Research Center, North Carolina State University, Raleigh. His research interests span the computational modeling of narrative, the use of artificial intelligence techniques within computer games, and virtual worlds.

Dr. Young is a senior member of the Association for Computing Machinery as well as a member of Association for the Advancement of Artificial Intelligence and of the International Game Developers’ Association.