ABSTRACT

JHALA, ARNAV H. Cinematic Discourse Generation. (Under the direction of Professor R. Michael Young).

Narrative is one of the fundamental ways in which humans organize information. 3D virtual environments provide a compelling new medium for creating and sharing narratives. In pre-rendered virtual environments like animated movies, directors communicate complex narratives by carefully constructing them shot-by-shot. To do this, a film's director exploits the viewer's familiarity with narrative patterns and cinematic idioms to effectively convey a structured story. In real-time environments like games and training simulations, however, a system has much less control over the stories that need to be told, since often novel stories are constructed on demand and tailored to a specific session or user's needs. In many contexts, stories are built not solely by the system, but collaboratively with many users whose choices for action contribute to the construction of unanticipated narrative structure.

In the past, intelligent cinematography systems have been developed to automatically record the actions of users within a virtual world and then to construct coherent visualizations that communicate these action sequences. While these systems generate coherent visualizations, they do not attempt to address the careful construction of narrative discourse based on established and identifiable patterns of narrative communication. Current automated camera systems take into account local coherence of shots and transitions but do not address the rhetorical coherence of the communication across multiple shots.

I describe an end-to-end camera planning system - Darshak - that constructs visual narrative discourse of a given story in a 3D virtual environment. Darshak uses a hierarchical partial order causal link planning algorithm to generate narrative plans that contain both story and camera actions. Dramatic situation patterns commonly used by writers of fictional narratives and endorsed by narrative theorists 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 shots and transitions are defined in this plan-based framework as execution primitives.

Representation of narrative discourse as a hierarchical plan structure enables us to utilize 1) the hierarchical nature of narrative patterns and film idioms through the hierarchy in decompositional plan operators, and 2) explicit representation of causal motivation for selection of shots through causal links. I present an empirical evaluation of the algorithm, based on cognitive metrics, for three properties of cinematic discourse: Saliency, Coherence and Temporal Consistency.

Cinematic Discourse Generation

by
Arnav Harish Jhala

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fullfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2009

APPROVED BY:

Dr. James Lester	Prof. Timothy Buie
Dr. R. Michael Young Chair of Advisory Committee	Dr. Jon Doyle

DEDICATION

TO
my parents
and
Shikha
as we embark on our
journey to parenthood

BIOGRAPHY

Arnav Harish Jhala was born on a cold January evening in the bustling city of Amdavad in India. He graduated from GLS high school in Amdavad in 1997 and moved on to LD College of Engineering where he majored in Computer Engineering. The internship at the Indian Space Research Organization was the highlight of his undergraduate education. He parted ways with his favorite city in 2001 after 21 great years for a land far far away. As a graduate student, he joined the Liquid Narrative Group, where his advisor, Dr. R Michael Young and his fellow graduate students kept him interested in challenging research topics spanning several research areas. Along the way he received his Master's degree in Computer Science in 2004 and took several opportunities to venture out to experience the "real" world through internships, first at Blackbaud, then at the Institute for Creative Technologies at USC, and finally at Virtual Heroes, Inc. Arnav is interested in pursuing research on computational models of filmmaking and developing AI for narrative-based learning environments.

ACKNOWLEDGMENTS

> The mediocre teacher tells. The good teacher explains. The superior teacher demonstrates. The great teacher inspires.

This dissertation is the fruit of inspiration drawn from many teachers and mentors. I am fortunate to have found my advisor Dr. R. Michael Young who has patiently guided me through all my shortcomings and mistakes to bring me to a level I had never imagined I would reach. His mentorship goes beyond teaching and research. I would always strive to be a teacher, mentor, and researcher like him. He has given me the tools, advice, and inspiration to succeed in all aspects of my future career, and should I succeed, it will be solely due to his mentorship. My committee members Prof. Tim Buie, Dr. Jon Doyle, and Dr. James Lester have been an invaluable source of knowledge, encouragement, and support. I have found conversations with Dr. James Lester stimulating and have enjoyed taking advanced classes with him. I thank him for his support and advice at various stages during this long journey. Special thanks go to Dr. William Bares for giving me valuable feedback on this research.

My excursions outside of the safe haven of the Liquid Narrative Lab have broadened my vision about the significance of my research. I am indebted to Drs. Andrew Gordon and Michael van Lent for giving me the opportunity to work at one of the finest research institutes in the world in the field of interactive technologies. They continue to be sources of support and valuable advice.

Many friends and colleagues at NCSU have provided an excellent social environment for easing the pressures of this dissertation. I would like to thank members of the LiquidNarrative and Intellimedia labs, in no particular order, Dr. Mark Riedl, David Christian, Justin Harris, Dr. Yun-Gyung (Yuna) Cheong, James Niehaus, Jim Thomas, Byung-Chull(Leo) Bae, Dr. Scott McQuiggan, Dr. Bradford Mott, Eunyoung Ha, Joe Winegarden, and all the Mimites and Zocalites for making the long stays in the lab bearable.

I am lucky to have friends like Sandeep Kachru, Tadatmya Vaishnav, Dr. Jaydev Jani, Dr. Nirmit Desai, Harshvardhan Joshi, Ajay Dudani, Kunal Shah, Dr. Kurang Mehta, the cyberhunks gang, and the shaitan gang who have always been around for me and have provided constant encouragement in tough times. My cousins have always maintained an academically competitive environment within the family and it has been fun growing up with them. I missed them all during these years but the fond memories kept me going.

Finally, my parents, Harish and Kapila Jhala for providing a healthy childhood and nurturing exceptional values without which this day would never have come. I am deeply grateful to Shikha for her love and support especially during the last few months while carrying our baby.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

> ”I have defined a *motivated* camera in relation to its narrative function within a film’s articulation of time, in relation to elements of classical narrative causality and in relation to a narrative sequence of changing shot scales.”
>
> *- E. Branigan, Projecting a Camera, 2006*

Over the last decade, advances in graphics hardware and the popularity of video games have led to an increasing number of applications for entertainment and education that utilize the immersive nature of such environments. With the availability of affordable processing power these environments have become very complex. For the user of such an environment, a virtual camera is a window into the environment’s virtual world. Far from being a simple, static viewport into the world, however, a camera is a powerful communicative tool that visually conveys information to a viewer. Film directors and cinematographers, for instance, have successfully used cameras to tell immersive stories that appeal to vast audiences. Many advances in the design of systems that communicate effectively within 3D virtual environments have focused on the development of communicative elements that operate within the virtual world via conventional means such as characters’ natural language dialog or the coordination of gaze, gesture or other aspects of agent embodiment. These approaches ignore, however, what cinematographers and movie-goers have long known: that the camera is itself a powerful communicative tool. This is also true for virtual worlds oriented towards the creation of narrative experiences (e.g., computer games, educational and training simulations). The very way that the camera is manipulated in its use conveys information about the graphical world beyond the communication of the objects and events perceived through it.

Film directors and cinematographers have developed the art of telling stories through a camera. The language of film has its own syntax and semantics that have evolved over the years. While camera control in interactive virtual worlds does not have the cinematic quality of films, I argue that certain strategies for camera placement in virtual environments can be borrowed from established conventions in film.

To illustrate the relationship between actions happening in a 3D environment and camera actions, and to motivate the representational requirements of an automated system for generating such camera actions, consider the following part of the climax sequence from the movie *Rope* [1] by Alfred Hitchcock. The scene contains 3 characters in a tense conversation. This sequence consists of a series of speech and movement actions of the characters. An annotated script for the scene is shown below. I will use the name C_i for camera actions and S_i, M_i and R_i for speak, move and react actions respectively.

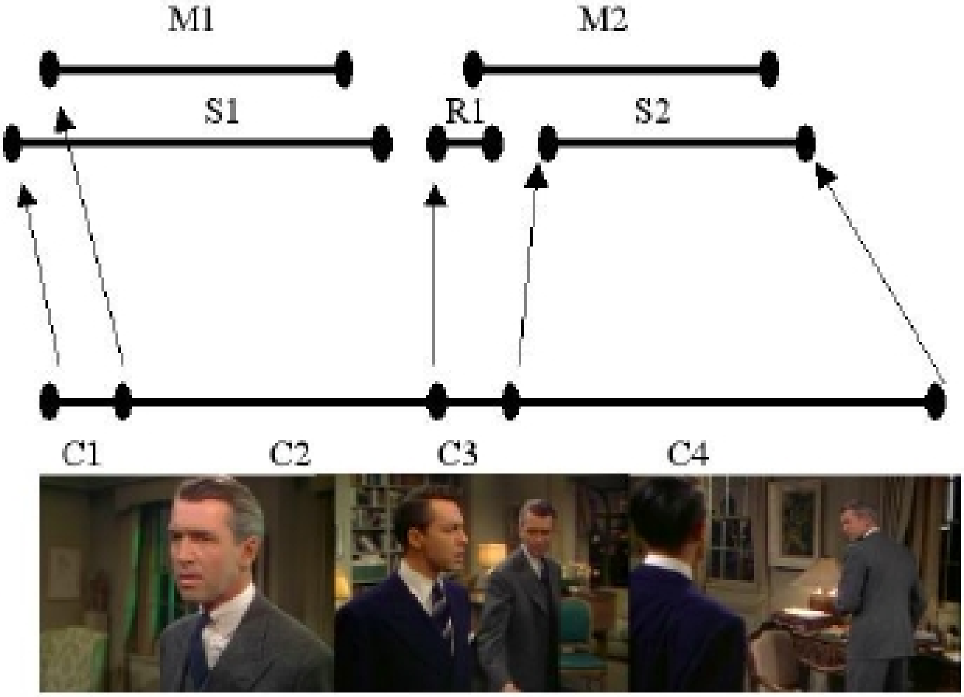


Figure 1.1: Relationship between camera actions and story world actions. The arrows indicate the start times and end-times of camera actions.

- **M_1**. Rupert walks away from the table, shocked.
- **S_1**. Rupert speaks as he walks towards Brandon.
- **R_1**. Brandon reacts with a surprised look.
- **M_2**. Rupert walks past Brandon.
- **S_2**. Rupert speaks as he walks past Brandon.
- **R_2**. Brandon is nervous and fearful in reaction to Rupert's speech.

As shown in Figure 1.1, we can see that the camera action C_1 maintains a three-quarter close-shot of Rupert as he speaks until the start of M_1. Action C_2's start time matches with the start of M_1 when the camera zooms out and tracks Rupert as he walks away from the desk. The following schedule was used to film the scene in the movie. We use the vocabulary proposed by Arijon [2] for describing the shots in this dissertation.

Schedule 1

- C_1. Close-Shot of Rupert who is shocked.
- C_2. Zoom out to Medium-Shot and track relative Rupert as he walks away from the desk.
- C_3. External Reverse shot of Rupert as he speaks to Brandon.
- C4. Close 2-shot to film Brandon's reaction.

There are many different ways of filming the same sequence of actions by using a variety of shot types. An alternate schedule for the camera actions for filming the same conversation could be the following.

Schedule 2

- C_1. Long 2-shot of both characters starting from the start of action $S_1$to the end of S_2.

The selection of a strategy for a particular sequence depends on the contextual parameters of the story as well as the physical setting in which the story is unfolding. For example, in Schedule 1 the selection of action C_1, a close up shot, is motivated by the intensity of emotion of the character. Maintaining low tempo can be achieved by selecting actions with fewer cuts or longer durations. Thus a zoom action (a shot with longer duration) is chosen over a direct cut to a Medium-Shot in C_2 from a Close-Shot in C_1.

The construction of an effective schedule depends on the rhetorical elements in the story. For instance, Schedule 1 would typically be chosen over Schedule 2 when the interpersonal relationship of the characters is not intimate or when the speech acts are not causally important to the other parts of the story [3]. Graphical environments in interactive games and simulations have more dynamic physical configurations compared to films due to the movement of players, characters and objects. A camera controller for these environments should be able to frame the shots in a cinematically acceptable way by taking into account the current physical state of the environments' virtual world. For instance, in C_1 (Figure 1.1) a cinematic shot is generated by framing the character on three-quarters right of the screen. In a dynamic environment such as a game, movement of a character in actions C_3 and C_4 could be unpredictable; as a result a camera controller would need to pay attention to the shot dynamically in order to correct composition.

This type of dynamic, intelligent camera control for narrative oriented virtual worlds involves decision-making at three different levels: choosing cinematic geometric composition, choosing the best camera parameters to convey the narrative context, and choosing shots and transitions to manipulate the viewer's mental states while maintaining the rhetorical coherence of the underlying narrative. Existing geometric composition algorithms [4, 5, 6, 7] serve as tools for placement of the camera given viewing and framing constraints. However, these algorithms provide limited automated constraint manipulation based on the underlying context of the narrative. Techniques that use geometric constraints also do not capture higher level cognitive processing across multiple shots in the viewer's mental model, such as discourse-level relationships between shots. This work addresses the high level reasoning mechanism for generating visual discourse and its integration with the underlying execution environment to produce visual discourse of cinematic quality.

1.1 Narrative and Narrative Discourse

Prince [8] defines narrative as a recounting of a sequence of events. The recounting of a sequence of events, in itself, may not be interesting. Only certain narratives actually tell an interesting story. These narratives have certain properties that distinguish them from other form or narratives, like a list of random events. Specifically, story-based narratives have a plot. A plot is the list of important events that form the outline of a story, a recounting of the order of events in time that follows a certain pattern [9]. Narratology is the study of such story-based narratives in media such as text [10] and movies [11]. Narratologists typically divide the narrative into three levels of interpretation. These three levels are:

1. *Fabula*: *Fabula* includes the story world, with all content, characters, actions, events, and settings needed to describe the rich setting of the story. Fabula includes the main events in a narrative that make up the general theme and influence the content of the narrative. Events in the fabula occur in the same chronological order in which they occured in the story.
2. *Sjŭzet*: *Sjŭzet* is a collection of events drawn from the fabula, selected and given an ordering by the author for communication to the viewer. Not all events from the fabula are necessarily included in the sjŭzet, for instance, when their inclusion would make the story overly detailed or when their presence could be readily inferred by the viewer. The ordering of events in the sjŭzet might differ from the ordering of those events in the fabula, for instance, when an author makes use of flashback or certain types of foreshadowing.
3. Realization Medium: The surface-level communicative form, such as words and sentences in the textual medium, that the narration takes.

The classification of narrative into these three levels is somewhat artificial. Authors often do not use three separate, clearly defined processes to create each of these levels. The construction of the *sjŭzet* often influences the requirements of the structure of the *fabula*, prompting revisions to both structures. Specific properties of the medium also lead to corresponding changes to the *sjŭzet* level. In my work, however, I employ a bipartite representation of narrative, the *fabula* part, described in the previous paragraph, and the discourse part that represents a combination of techniques for construction of the *sjŭzet* and realization through the visual medium. This is a purely functional classification due to the availability of algorithms that reason about the *fabula* independently from the discourse part.

1.2 Problem Description

Narrative discourse is a recounting of events occurring in a story world through a communicative medium (in my case a 3D graphical world). As described above, narratives are about communicating transformations in (fictional) world states into a final state through a sequence of events. Certain narratives are called ”stories” while others are not. These narratives, that have stories, contain properties like settings, objects, characters, and characters' beliefs, desires, plans, and goals in the story world. Film directors and master storytellers take advantage of these properties of stories and exploit them to craft interesting recountings or telling for these stories. Cinematics crafted by experts through manipulation of these properties are effective when viewers find the stories communicated by them coherent and the visualizations aesthetically pleasing.

In order to produce a system that generates effective cinematics, I define three properties of the narrative viz. saliency, coherence, and temporal consistency that are not addressed by previous approaches in generating automated visualizations. I claim that these properties are central to the comprehension of visual narrative. Chapter 3 of this dissertation defines an approach to the generation of visual narrative directly addressing these properties and Chapter 4 provides an evaluation that supports the claim that Darshak produces effective cinematics by leveraging their representation. The approach presented in this dissertation is informed by established cognitive models of narrative comprehension. Cognitive psychologists have determined that the ability of an audience to comprehend a narrative is strongly correlated with the causal structure of the story [12, 13, 14]. Story comprehension is also dependent on the audience's perception of the causal relationships between story events. In the visual medium, audience perception is influenced by the quality of the cinematic and the choice of camera properties to visualize the story events.

Selection of salient elements: Salient elements in a discourse are elements from the story (actions, events, characters, objects, relationships) that are relevant to inferences needed for comprehension. When stories are narrated through any medium, it is the narrator's responsibility to utilize the properties of the medium to maintain the engagement of the audience by providing them with the relevant information at the right times during the telling. It is important for discourse in any medium to maintain focus of the audience to salient elements of the story. Having extraneous elements, not part of the causal chain of the story, or leaving out necessary details of the story can deter the audience from enjoying the narrative experience. Any narrative discourse generation algorithm must make either explicit or implicit choices about content from the story that affect salience and comprehension.

Plot coherence: Plot coherence can be described as the perception by the audience that the main events of a story are causally relevant to the outcome of the story. This definition of plot coherence is taken from the Fabulist story generation system [15]. Plot coherence relates specifically to the the perceived understanding of the causal relationships between events that a viewer constructs during narrative comprehension. If events narrated to an audience seem unrelated to the story's final outcome then the audience may make incorrect inferences about the narrative and the narrative discourse may fail to adhere to the author's communicative goals.

Temporal consistency: Temporal consistency, in my work, refers to the consistency in the timing of the camera positions and movement in relation to the events that are happening in the virtual world. Temporal consistency is important as it is closely linked to the established conventions of cinematography that viewers can readily identify. Any inconsistencies in the timing or placement of the camera affects the communicative meaning of the shot perceived by the audience. For example, moving the camera away too early while filming an action leaves the viewer confused about the completion of the action and may introduce an unintended inference in the mind of the viewer. Cutting away from an action prior to the action's completion raises a question in the mind of the viewer about the action's success or failure. Viewers have grown familiar with many cinematic idioms that are routinely used to indicate directors' specific communicative intentions. It is thus important for the intelligent camera planning system, Darshak, described in this dissertation to be able to reason about appropriate temporal relationships between camera movement and story events executing in a game environment.

For this dissertation, I define the specific problem statement in relation to the properties of visual discourse that were mentioned earlier. Specifically, the Darshak system is designed, developed, and evaluated as a solution to the following problem:

> **Problem Statement:** Given a story, in the form of parameterized set of actions with causal and temporal relationships between them, and a set of communicative goals, construct a schedule of communicative shots and create a visualization of the story such that it is coherent, comprehensible, and follows acceptable

cinematic conventions.

There are several technical challenges that arise during the development of a solution for this problem. It is appropriate, at this point, to briefly introduce the reader to previous approaches to camera control. This discussion motivates the challenges faced by automated camera control system. In the later section, these general challenges as well as specific challenges that arise from my approach are discussed.

1.3 Automated Camera Control Research

The earliest approaches to the problem of automated camera control [16, 17] focused on automatically detecting and avoiding occlusions when placing a camera in a 3D world so as to provide a clear view of the world's relevant graphical elements. These approaches used constraint-based techniques for specification of salient graphical elements and their presentation priorities and computing the best camera positions and viewing angles. These early approaches tried to find answers to the question of *where* to place the camera.

With advances in narrative-oriented graphical worlds and their increasing use in training simulations and other pedagogically oriented applications, the question of *how* to place the camera arose in the research community. This was addressed by the Virtual Cinematographer [18] system by formalizing idioms - stereotypical ways to film shots that were identified by cinematographers over the years from their filmmaking experiences. Other efforts used agent-based approaches, for instance, modeling the camera as an invisible creature [19] inside a 3D world that responded to the emotion of the scene and its characters to select an appropriate camera shot.

More recent approaches to camera control have begun addressing the reasoning 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 the challenges in the natural language discourse generation community for choice of appropriate linguistic communicative actions. In this community, theories of text discourse structures [20, 21] have been successfully used as the basis for the generation of coherent natural language discourse [22, 23, 21]. Integrating these theories with research on camera control has led to implementation of multimodal systems [24] that use the same underlying rhetorical structures to generate discourse in the form of natural language as well as directives for the camera.

1.4 Contributions

This dissertation describes a novel approach towards representation of and reasoning about cinematic discourse. The system that implements this approach for cinematic discourse generation is called Darshak. Darshak is an end-to-end system for generating movies from stories. Darshak extends existing research in intelligent camera control further through the following contributions:

- **Geometric Camera Control:** Darshak contains an implementation of film shots that are represented by a collection of constraints. These shots are pre-defined according to standard cinematic composition constraints but are parameterized. This work extends prior approaches to modular constraint based camera control by associating camera modules with planning operators with an explicit formalization of their effects on the cognitive model of a viewer.
- **Discourse Generation:** Darshak is a system that generates visual communicative acts by utilizing the same higher level planning representations that are used by natural language discourse generators. Darshak further extends the representation of communicative actions with an explicit representation of durative communicative actions and temporal constraints between communicative actions and between the occurrence of the physical actions they convey.
- **Intelligent Storytelling:** Most work on intelligent story generation systems has focused on story structure and the friction between user control and author control of stories. Less emphasis has been placed on the visual communicative effects of the story that is presented to the user. Darshak system specifically contains a representation of the effects of visuals on viewers.
- **Evaluation of Story Generation Systems:** Text-based story generation systems like Author [25] and Fabulist [15] have utilized cognitive evaluations of the stories

they produce. Automatically generated stories have not yet been evaluated in the visual medium. This work is a step towards providing methodologies for evaluating automatically generated stories in a new medium.

1.5 Reader's Guide

It is not necessary to read the entire dissertation sequentially. The reader is free to explore the chapters that interest them the most. This section describes the overview of what readers can expect from the chapters in this document.

Chapter 2 is a discussion of related work. The related work for this research is spread across multiple disciplines. The first part of Chapter 2 describes in detail prior approaches to automated camera control research, mainly from the graphics community. Other sections introduce readers to concepts from the fields of discourse generation, planning algorithms, narrative theory, and film theory. The algorithm implemented in Darshak is a discourse planning algorithm that incorporates a representation of film idioms and narrative patterns in a plan operator formalism.

Chapter 3 discusses the core theory and implementation of the various parts of the Darshak system. First, a formalization of narrative patterns and cinematic idioms as plan operators is provided. This formalization is motivated by the hierarchical nature of discourse in general, and visual discourse in particular. Then, the extensions to traditional discourse planning algorithms are discussed that are unique to planning narrative-oriented visual discourse and the revised planning algorithm is described in detail with a complete example demonstrating the working of the system. Finally, the overall software architecture and the execution environment of the Darshak system and its relationship to the Zocalo service-oriented storytelling architecture are described. This chapter also covers the detailed implementation of procedural camera control classes within the game engine that correspond to plan operators and describes the translation of declarative operators to procedural code objects.

Chpater 4 describes an empirical evaluation of the Darshak system, focusing on the efficacy of the discourse planning algorithm. The purpose of the evaluation is to determine the effect that visualizations generated by the system have upon viewer's comprehension of the story and upon their perception of the stories' coherence. In addition to the quantitative metrics, several qualitative responses from the viewers are presented.

Chpater 5 summarizes the research and reiterates the main contributions of the work. It also presents a reflection on conclusions drawn from this work and provides an overview of future research directions.

Chapter 2

Related Work

The research problem addressed in this thesis and the solutions presented here draw upon various techniques in Computer Science, Film Theory, Narrative Theory, Communication Theory, and Psychology. In my approach, camera control in a virtual environment to tell a story involves: Determination of important events in the story (Narrative Theory), Temporal and Spatial organization of events for telling (Film Theory), selection of appropriate conventions of communication (Linguistic Communication Theory), and intelligent determination of geometric parameters of the camera in the 3D virtual environment (Artificial Intelligence, Graphics). In this chapter, I attempt to describe in brief the techniques, theories and related systems that provide motivation of the various aspects of my solution.

2.1 Geometric Camera Placement

For any dynamic 3D graphical system, one of the fundamental operations is the manipulation of the system's viewport. Manipulation of graphical viewport is achieved by selection of numerical values for the 6 degrees of freedom for the camera, namely x, y, z, pitch, roll, yaw, and other parameters of the camera like the FOV(Field of View).

Many systems, like first person video games, give users complete control over the positioning of the camera. However, other systems, such as training simulations, cannot afford such a degree of user control. They require the user to view critical elements of the scene that satisfy the pedagogical goals of the system. User control of the camera in these contexts would allow the user to change the view in arbitary ways that might obscure or remove from view elements that are critical to satisfaction of pedagogical goals.

Christie et. al. [26] classify geometric camera placement approaches along two dimensions: expressivity and solving mechanism. Here expressivity relates to the ability of the language used for geometric specification to express a wide variety of camera placement parameters. Solving mechanisms refer to the mathematical approaches and algorithms applied to the camera planning problem. Solving mechanisms are further divided into various classes based on the specific techniques that are implemented. These classes are:

- **Algebraic systems** represent the problem in vector algebra and directly compute a solution;
- **Interactive systems** set up cameras in response to user interaction through an input device and provide immediate feedback to the user;
- **Reactive real-time systems** utilize motion-planning algorithms from robotics to reactively move cameras in dynamic environments;
- **Optimization and constraint-based systems** represent camera properties as constrains and optimize the objective function through a variety of solving processes that widely differ in their properties (e.g. incompleteness, softness, etc.)

Early research on geometric camera placement was motivated by interactive graphical applications like planetary motion simulation, and manipulation in CAD environments [27, 28]. These were approaches that focused on providing user feedback of their manipulation actions through camera movement and are examples of **interactive systems**. More recent approaches have focused on fully automated camera movement that is based on cinematic principles and executes in dynamic environments. Interactive systems and algebraic systems were good for static and less dynamic environments with significant amount of user control. Reactive systems and constraint-based systems developed as the need for more complex camera movement was felt in many applications like computer games[5] and pedagogical simulations[17]. Constraint-based and interactive systems are most relevant to the approach. Some of the relevant techniques are described in the following sections.

2.1.1 Geometric Composition of Camera Shots

CamDroid

CamDroid[5], developed by Drucker and Zelter, introduced the notion of camera modules for procedural camera control. They established a constraint based framework for implementation of camera control in which desired geometric compositions were expressed in the form of constraint lists with priorities given to each constraint. The constraints were used to restrict the possible solution space and to establish the position values as well as the degrees of freedom of the camera. Examples of such constraints are: constraints that restrict the camera to be at a certain height(restrict Y), maintain the camera's *up* vector to be the world's *up* vector(restrict *roll*). This system also used a modularized architecture to encapsulate different styles of interaction. It had a library of preset constraints for standard shots like the over-the-shoulder shot that is commonly used in conversation sequences.

CamDroid included a general interpreter for running pre-specified scripts and managing user intervention. Commands for manipulating camera modules were specified in TCL with the interface created in TK, a built-in rendering sub-system on the SGI graphics engine, an object database for a particular environment, and the camera modules that were pre-specified by the user for each domain. As an example of the type of constraints handled by CamDroid the **Over the Shoulder** shot type that was part of the conversation illustration in Drucker's work is specified using the constraints:

- Height of the character facing the view should be approximately 1/2 the size of the frame.
- The person facing the view should be about the 2/3 line on the screen.
- The camera should be aligned with the world up axis
- The field of view should be between 20 and 60 degrees.

While this approach produced acceptable pre-computed camera parameter solutions for constraint satisfaction, it did not handle failed constraints or real-time resolution of constraints and was subject to local minima failures. It also used pre-scripted directives and modules for execution. These issues were addressed by ConstraintCam [17].

ConstraintCam

The ConstraintCam system [4, 17] is a real-time visualization planning tool that generates camera shots to satisfy the user-specified goals in the form of geometric composition constraints. The design guidelines that ConstraintCam is based upon are:

- *Arbitary Viewing Goals:* Viewers can update the constraint set at any time or request alternate views of any subset of initially specified constraints.
- *Environmental Complexity:* It supports planning for solutions in complex environments by taking into consideration high geometric complexity, holes, or transparent surfaces.
- *World Non-Interference:* Limited support for world manipulation to artificially create good compositions. For example, the system can opt to move an object that is out of position in the shot composition.
- *Failure Handling:* Systematic methods such as the ConstraintCam solvers produce sub-optimal shots by relaxing constraints after taking into consideration their relative priority.

A version of ConstraintCam supports 15 types of constraints for each object. The constraint types have a general format with a range of allowable values, an optional optimal value, and a value indicating importance of the constraint. A few constraint definitions are provided here [29].

- **LOOK_AT_POINT** This constraint causes the camera to focus on a specified point.
- **OBJ_IN_FIELD_OF_VIEW** This constraint requires a primary object parameter and maintains full visibility of the object's bounding volume within the field-of-view.
- **OBJ_OCCLUSION_MINIMIZE** A fraction of the allowable occlusion value can be specified. The primary object of attention cannot be occluded more than this allowable value. The fraction of actual occlusion is calculated using the ratio of pixels being occluded to the total number of pixels taken up by the unoccluded object on the screen at a particular field-of-view.

For a given situation, the solver computes the satisfaction value of each constraint to determine the quality of the solution. A cumulative constraint satisfaction rating is calculated from a weighted sum of the satisfaction measures of all individual constraints from the equation shown below.

$$S = \sum_{1}^{N} (P_i \times S_i),$$

Where,

S : overall satisfaction value

N : number of constraints

P_i : preferred value for the i^{th} constraint

S_i : satisfaction value for the i^{th} constraint

More recent work on constraint based systems[7, 6] is mainly geared towards optimization of constraint solution in real-time, and incorporating aesthetic composition of shots based on the underlying context of the story and the geometry of the world.

Film Idioms

One of the main drawbacks of geometric constraint solving systems is that they are tied to the underlying geometry and do not reflect the cinematic motivation for generation of constraints that is dependent on the context of an application. Several camera control architectures [18, 30] have been built to make use of the cinematic idioms for virtual cinematography.

- **The Virtual Cinematographer**

  The Virtual Cinematographer [18] introduced the ability for developers to define specific film idioms using a scripting language called Declarative Camera Control Language(DCCL). The Virtual Cinematographer program dynamically chose a shot sequence from a set of candidate idioms specified in DCCL. The major contribution of this work was to formalize the concept of film idioms and to apply the formalization in an interactive application. The virtual cinematographer takes as an input a specification of camera shots in terms of desired positions and movement of actors across the scene. A heuristic evaluator then ranks the candidate idioms on the basis of smooth transitions, crossing the line of interest, long fragments, and backward panning of the camera. While the system effectively considers these types of cinematic rules, it does not capture the mood or emotion in the scene while choosing an idiom, unlike the actual director and cinematographer creating a film.

- **FILM System**

  Amerson, Kime and Young [31] suggest a method of adapting high-level cinematographic directives to fit with the underlying graphical environment in real-time. They use a layered approach, with a director module that interprets action specification for a scene tree in the ontology of the FILM (Film Idiom Language and Model) language. Their system does a breadth first search to find the best node in the tree that satisfies the affective parameters of the specification. The director module then hands off the selected scene to a cinematographer module that binds the camera parameters by converting scene specification into geometric constraints. The use of this model allows for abstracting out the idiomatic knowledge from the graphical subsystem.

- **Conveying Emotion Through Cinematography**

  One model used to motivate camera moves, especially in narrative systems, is based on the ethologically-inspired structure of sensors, emotions, motivation and action-selection mechanisms. In systems that use this model, each character in a narrative exhibits the influence of emotions in its action-execution. This property of autonomous characters is exploited by the CameraCreature[19]. Tomlinson, Blumeberg et al. [19] have recently used expressive characters in this way to drive the cinematography module. This is an interesting approach as it captures the emotion of characters and the mood of the scene that is conveyed to the viewer in a cinematic manner.

Cinematographic Tools and User Modeling

As a result of the research in the field of cinematography, various applications have been developed that are built from the parts of systems mentioned in the previous sections.

- **UCam: User Modeling for Adaptive Virtual Environments**

  UCam [32] system uses a user model to specify the visualization preferences of a viewer.

The system then customizes camera movements in real-time based on dynamic user interaction with the environment. The UCam system has been evaluated through user studies and shows promise of providing effective camera control interfaces in adaptive virtual environments.

- **Longboard Storyboarding Tool**
  Jhala et. al. [33] describe an intelligent storyboarding tool that takes as an input abstract camera action specifications annotated by situational attributes. It translates these specifications into geometric constraints by using rules borrowed from Hollywood filmmakers to generate shot compositions. This system attempts to bridge the gap between the high-level semantics of a shot with respect to the underlying story actions and the optimal camera placement with respect to the geometric setting of a story's virtual set.

Generation of Video Documentaries from 2D Images

Callaway et. al. [24] describe a system for generating video documentaries from 2D images and narration in the form of audio clips. The approach uses a repository of annotated images and uses a set of cinematic idioms to generate constraints that are used in conjunction with the planned rhetorical actions for natural language, speech, and camera to generate a coherent sequence of documentary video for the given images. This is an interesting approach as it combines the concepts from many of the previous cinematography systems into one application.

2.1.2 Limitations of Intelligent Camera Placement Systems

In order to effectively construct a visualization that communicates a story occurring in a virtual environment, an abstraction of geometry is needed. Objects considered as geometric primitives provide incomplete information about the role of objects within a story in the virtual environment. Geometry based approaches face this obstacle as they do not incorporate a rich model of object and event abstraction. As most researchers in geometric camera placement point out, it is difficult to improve the quality of automated cinematography unless a representation of complex cinematic idioms is included in the system. The expressiveness of the system is related to the spatial and temporal context of the scenes and the communicative intent of the user. In this thesis I present a system that addresses these issues.

2.2 Overview of Computational Techniques Used in Story Generation

One of the motivations of the automated camera control research described in this dissertation is to provide a visualization system for automated story generation system based on planning techniques. The Darshak camera planning system relies on the rich representation of story elements for reasoning about the best visualizations of those stories. A rich computational model of story structure is necessary for Darshak to generate better visualizations.

Many automated story generation systems [34, 35, 36, 37, 38, 39, 15] have been developed over the years with several different computational representations of story structures. Among these different computational representations of the story, the plan-based representations of the story are most relevant to this work. Plan-based representations explicitly encode actions in the story as well as the causal and temporal relationships between story actions.

Given the large number of systems that can be characterized as story generation systems, it is beyond the scope of this dissertation to enumerate and describe all of them. The story generation system described here is the one that is directly related to the research on camera control described in later chapters.

2.2.1 Plan-based Computational Models of Narrative

Planning formalism provides a good computational model of stories for two reasons. First, the plan data structure (described in detail in the next section) is a good representation of the plot structure with actions and events that are part of the story and a temporal ordering on them. This representation is consistent with the definition of a plot (or fabula) given in narratology.

Several story generation systems [15, 38, 35] adopt the planning formalism for representing stories. In this section, I will describe Universe [35], I-Storytelling [38], and

Fabulist [15] in more detail. Darshak's algorithms operate on the representation dervied from the Fabulist system.

Fabulist

Fabulist [15] addresses the problem of story generation through a system that implements a model of dramatic authoring such that it generates coherent stories with believable characters. It takes an author-centric approach to accomplish coherence and character believability. Fabulist incorporates explicit representation of intentionality of characters to increase believability. This, combined with open-world planning, leads to a limited form of plot coherence termed as story coherence. Riedl [15] defines story coherent narrative as a narrative in which all events have meaning and relevance to the outcome of the story. Since a planner only adds actions into a plan that serve to achieve the goal, all actions in a story plan are part of some causal chain leading to the outcome of the story. Thus, story coherence comes about as a side-effect of planning.

Universe

The Universe system [35] utilizes a planning algorithms to generate open-ended stories. Open-ended stories do not have a specified outcome. The planning algorithm takes authorial goals on an episodic basis to generate plot fragments, similar to abstract operators in NOAH [40], for each individual episode in such open-ended narratives. As the plot-fragments are added to serve author's goals, they contribute sub-goals that are satisfied by a planner until a level is reached where the story can be told.

I-Storytelling

The I-Storytelling project [38] has autonomous characters, each with their own HTN (Hierarchical Task Network) planning algorithms, generate plans to satisfy their individual goals. The hierarchical task networks are hierarchies of sub-goals and actions. The HTN planner for each agent generates a plan by searching through AND/OR graphs corresponding to the HTN structure. The initialization of characters and the virtual world influences the outcomes of the story. In this representation, the way in which characters individually form goals and satisfy them is not affected by the emerging global plot of the story.

The next section gives an introduction to planning, specifically plan-space planning, as the planning representations and planning algorithms are central to the technical content of this dissertation.

2.2.2 Overview of Planning

Planning is a technique for generating novel action sequences in response to the specification of a desired state of affairs in the world. Typical planning problems are represented as having three parts to their specification: an initial state, a goal state and a set of operators that describe valid actions in the world. The process of solving a planning problem involves contructing a sequence of actions that, when executed starting in the world described by the initial state description, will successfully terminate after changing the world to be one described by the problem's goal state description.

There are many approaches to solving the planning problem. In this work, we use a planning algorithm that employs *refinement search* [41] to construct plans. Refinement search is a general characterization of the planning process as search through a space of plans. A refinement planning algorithm represents the space of plans that it searches using a directed graph. Nodes in the graph represent plans and arcs between nodes indicate that the the nodes at the ends of the arcs refinements of the nodes at the origins of the arts (that is, the plan associated with the second node is constructed by repairing some flaw present in the plan associated with the first node). In typical refinement search algorithms, the root node of the plan space graph is the empty plan containing just the initial state description and the list of goals that together specify the planning problem. Nodes in the interior of the graph correspond to partial plans and leaf nodes correspond to completed plans (solutions to the planning problem) or plans that cannot be further refined due to inconsistencies within the plans themselves.

A plan *operator*[42] is represented by:

1. A set of variables
2. A precondition list: A list of conditions that must hold at the time of the action's execution.
3. An effect list: A list of conditions asserted after successful completion of the action.

4. A set of constraints on the operator: A list of constraints specifying valid object bindings for the variables of the operator.

The planning algorithm instantiates operators that either meet the goals of the plan or enable actions that lead to the goals. Instantiated operators are added to the plan data structure as plan *steps*. A *binding constraint* is a tuple that of the form *(variable, object)* that constrains the binding of each variable in an instantiated step. The relationship between actions that establish preconditions of other actions is represented by *causal links*. When an action interferes with the effects of another action it is said to *threaten* the causal link. Threats are resolved by the planner either by explicitly adding an *ordering* constraint or by adding an action that resolves the threat. An *ordering* constraint is tuple of the form *(prior, latter)* where *prior* is a step in the plan that occurs before *latter*.

Formally,

Definition 1 *(***Plan***) A plan P is a tuple* $\langle S, C\rangle$ *where S is a set of steps and C is a constraint tuple of sets of constraints on S. Minimally, C contains a set of ordering constraints O that define a partial temporal ordering on the execution order of steps in S.*

A planning problem Given a planning problem the ultimate aim of the planning algorithm is to come up with a sequence of ground operators such that by application of these from the initial state, the goal state is attained and each step is consistent with its constraints.

Definition 2 *(***Planning Problem***) A planning problem is a three-tuple* $\langle P_0, \Lambda, \Delta\rangle$ *where* P_0 *is a plan specifying the initial and the goal states,* Λ *is the planning problems set of action operator definitions and* Δ *is the set of decomposition operator definitions.*

A ranking function defines a pre-order on the plans generated by the planning algorithm for a particular planning problem. This function is used to guide the expansion of nodes at the fringe of the plan space during refinement search. The system designer can choose to use a custom ranking function that operates on the plan data-structure and gives a numeric ranking to each plan. A ranking function could, for instance, give a higher numeric ranking to plans that contain more abstract steps than primitive steps. This can be implemented by having the ranking function count the number of abstract steps in each plan data structure and assign this number as a numeric value to rank the plan.

Definition 3 *(***Ranking Function***) For any set of partial plans for a particular planning problem P_{pp}, a plan ranking function f:$P_{pp} \rightarrow N$ defines a pre-order on the plans in P_{pp}.*

In this dissertation I develop a modified version of the DPOCL (Decompositional Partial Order Causal Link planning) [22] algorithm to represent the sequences of shots for viewing a narrative, also in the form of a Longbow plan data structure. Longbow uses a least commitment planning approach to generate plans for achieving the goals from an initial state of the world. The resulting plan has a list of partially ordered steps with causal links. Causal links establish the ordering of steps in cases where the precondition of one step is established by another step in the plan. This imposes a partial ordering constraint on the steps and at the same time, from a discourse theoretic perspective, captures the rhetorical structure for communicative acts represented as plan operators.

Definition 4 *(***Longbow Plan***) A Longbow plan is a tuple $\langle S, B, O, L_C, L_D \rangle$ where S is a set of steps, B is a set of binding constraints on free variables in S, O is the set of ordering constraints on steps in S, L_C is the set of causal links between steps in S and L_D is the set of decomposition links among steps in S.*

Definition 5 *(***Causal Link***) A causal link is defined as CL= $\langle S_i, S_j, C \rangle$ where S_i is the step whose effect establishes a precondition C of step S_j.*

DPOCL uses hierarchical planning in addition to causal planning. Hierarchical planners support action decompositions for abstract action specifications. A hierarchical approach has a number of potential benefits. First, it may lead to improved performance due to the reduction in amount of search needed. Second, the hierarchical action representation may match human designers' notions of actions in the domain, making encoding of actions more straightforward. Finally, a hierachical approach facilitates interleaved planning and execution by allowing the specification of details of the plan that will occur later to be deferred while elements of the plan that are ready to execute near its beginning to be sent to an execution system. In my work, the hierarchical approach matches well with abstract structures seen in story and cinematic patterns.

Definition 6 *(***Action Schemata***) An action schemata is a tuple $\langle A, V, P, E, B \rangle$ where A is the action type, V is a list of free variables, P is a set of preconditions for the action,*

E is the set of effects for the action and B is the set of binding constraints on the variables in V.

The agent in the world can directly execute primitive actions. Abstract actions reflect the intentions of the designer to execute high-level goals that are realized by multiple related primitive actions, the effects of which combine to realize the high level goal.

Definition 7 *(***Primitive/Composite Actions***) The set of actions Λ_{prim}, in a given action schemata Λ, whose members are all primitive actions. All non-primitive actions are composite actions.*

Table 2.1: Definition of a primitive action

```
(define (action DESCRIBE)
    : parameters (?V, ?X)
    : primitive t
    : description Primitive action describe object
    : precondition NIL
    : effect ((BEL(?V ?X))
    : constraints NIL)
```

Definition 8 *(***Decomposition Link***) A decomposition link is a tuple $\langle S, S_i, S_j, S_f, S_s \rangle$ where s is a composite step, S_i is the initial step of the decomposition, S_f is the final step if the decomposition and S_s is the list of all the interior siblings of the decomposition.*

Longbow is a decompositional partial order causal link planning system based on the DPOCL algorithm[22] that has been used in the generation of natural language discourse (e.g., multi-sentential texts such as paragraphs of instructional texts). In this work, I based the planning algorithm used on Longbow's implementation. Details of my planning algorithm are discussed in the next chapter.

2.2.3 Linguistic Discourse Structure

Most research in computational cinematography has focused on the denotative meaning of shots, that is, the meaning of shots determined by the objects, actions, and locations appearing in the shot. The selection of shots is governed by the actions taking place in the world and the realization is motivated by geometric accuracy of camera placement. There is a need for camera planners to consider the connotative meaning of sequences of shots that clearly identify the rhetorical relationships between adjacent shots and shot sequences. Researchers [43, 24] draw an analogy between shots and sequences in cinematography and textual sentences in multi-sentence discourse in natural language. The theory of discourse structure due to Grosz and Sidner [20] takes into account the correlation of discourse segments while considering the beliefs, intentions and attentional state of participants. At a higher level of analysis of text, Rhetorical Structure Theory (RST) [21] seeks to describe the structure of discourse through the relations that hold between parts of text, and the schemas identified by abstract patterns of small spans of text. These discourse structures have been used to build discourse planning systems that generate natural language speech acts for communication. Systems like FILM [43] employ similar techniques for reasoning about camera shots as communicative acts.

Theories of Discourse Structure

Grosz and Sidner's theory of linguistic discourse structure[20] is a composite of three interacting parts: the linguistic structure, the intentional structure and the attentional state. The linguistic structure has as basic elements utterances while the intentional structure comprises a small number of relationships between these utterances. The Attentional Structure contains information about the objects, properties, relations and discourse intentions relevant at any given point. At the intentional level, discourse consists of segments that serve the purpose of communicating the intentions of the participants. The Discourse Purpose (DP) is the overall goal of discourse. Individual segments also have Discourse Segment Purposes (DSP) that contribute to achieving the overall DP. There are two structural relations that hold between the DSP: dominance, when DSP1 partly satisfies DSP2, and satisfaction-precedence, when DSP1 has to be necessarily satisfied for successfully satisfying DSP2. The attentional structure is modeled by considering focus spaces of the reader/viewer/listener consisting the properties, objects, relations and DSP that are salient for a particular discourse segment. These focus spaces are stacked to reflect the shift of attention across different discourse segments.

Rhetorical Structure Theory[21] captures the intentional and the informational relationships between elements of within a discourse. RST schemas are defined in terms of relations and they specify how text can occur coherently and relate to other text segments. In RST the primary text segment is identified as a nucleus if it satisfies certain intentional and informational constraints. Other text segments that satisfy the constraints on the satellite schemas are then identified as satellite segments. RST identifies five schema types that show how text segments are related to one another. Following relations are representatives of each of the five schema types: Circumstance, Contrast, Joint, Motivation/Enablement, and Sequence/Sequence.

Planning for Discourse Generation

Discourse generation systems use the theory of discourse structure discussed in the previous section to plan natural language text/utterances. Maybury [23], in his TEXPLAN planner, formalizes communicative acts as plan operators and generates multi-sentential text using a hierarchical planner. TEXPLAN uses rhetorical predicates to give a semantic classification of utterances in natural language. Surface speech acts are also classified such that they guide the selection of appropriate sentence structure using the underlying rhetoric propositional content.

The discourse planner used to generate advisory dialogues by Moore and Paris [44] generates plans that capture both the intentional goals of the speaker and the rhetorical means to achieve them. This planner is based on Rhetorical Structure theory discussed in the previous section. A selection heuristic for capturing the rhetorical means to achieve the intentional goals is used to select between many different rhetorical strategies for achieving a given communicative goal. Their planner uses the selection heuristic to select the most appropriate plan operators for conveying a given rhetorical strategy. The plan operator, in addition to the effects representing the intentional goals, also has a nucleus that represents the main topic either as a primitive speech act or intentional/rhetoric goal, which is further expanded, and a satellite that represents optional sub-goals for conveying additional information to support the nucleus.

The Pauline system [45] generates stylistically appropriate text from a single representation under various settings that model the pragmatic circumstances of the desired output. Pauline adds communicative actions through manipulation of intermediate rhetoric goals. The combination of these intermediate rhetoric goals is the way a speaker's pragmatic goals index to their stylistic opinion. This work is particularly interesting as cinematographic style is driven by the context in which the actions take place within the narrative.

The Mimesis FILM system[43] extends a natural language discourse planner, Longbow, to plan visual discourse segments. Longbow [22] implements a decompositional partial order causal link planning algorithm (DPOCL) [22]. In partial order causal link planners, a plan is represented as a set of partially ordered steps. Steps in the plan are linked through causal links in a way that steps at the source of the link establish some precondition(s) of the steps at the other end of the link. Causal links reflect the relationships between discourse segments. Longbow uses causal links to model the hearer combining the utterance with his/her beliefs that change with subsequent utterances in the discourse.

2.3 Dramatic Patterns in Narrative Theory

The computational narrative structure of the stories in this dissertation are drawn from the idea of dramatic patterns occurring in popular narratives. Georges Polti, in his book titled *The Thirty-Six Dramatic Situations*[46], describes 36 types of situations that he claims are commonly found in most dramatic literature. These are plots and subplots that are strung together to form the full story. Different story genres can be constructed using Polti's patterns. Each pattern, at the discourse level, imposes constraints on the story for inclusion and exclusion of certain events and actions or character types. Tobias [47] argues that viewers have to identify and interpret these patterns. Hence, it is a director's job to establish each pattern, maintain it and provide a resolution required by each pattern to meet or break viewer's expectations. For example, gunfight scenes are commonly expected by viewers in a Western movie. Each character in the movie provides an opportunity for the director to set up the story leading to the gunfight to make it interesting to the viewer. Some of the patterns that are formalized into plan operators are described below.

Another pattern, that is related to Deliverance, is *Supplication.* Converse to Deliverance, in which an unexpected protecter-of her own accord-comes suddenly to the rescue of the distressed, here the unfortunate appeals to an undecided power. This definition unlike deliverance does not have a specific constraint on the rescuing power or authority. In this case there are several realizations of Supplication that can occur within the discourse

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

Participants: Three (P_a, P_b, P_c)
Constraints on Participants:
Pa is a weak person. Pb is a strong person. Pc is established as a weak person
who overcomes the weakness to rescue the suppressed person.
Relationship with other strategies
The character personalities are explicitly established before application of this pattern.
Reversal of fortune is a sub-pattern that requires cinematically communicating
the action that leads to the rescuer overcoming the weakness.

Table 2.2: Operator definition of *deliverance* pattern

I. Supplication (Persecutor, Suppliant, PowerInAuthority)

Persecutor : Punisher, Harrasser.
Suppliant: sinner seeking forgiveness

Scenarios
A: The power whose decision is awaited.
Shall he yield to the menaces of the persecutor?
B: The PowerInAuthority is an attribute of the Persecutor,
like a weapon in hand. Shall anger or pity determine his course?
C: The third person becomes an intercessor for the persecuted.
There could be an additional person who acts as an intercessor.

based on the context and available resources in the story world. Here is a description of the variations of the Supplication pattern. One example of this pattern would be when the rescue takes place due to a divine intervention in response to the protagonist offering prayer. In this case the protagonist does not take active part in the rescue. Note that the overall flow of the story and the events that lead up to variations of deliverance or supplication patterns would be quite different when they are carefully constructed to lead up to the actual communicative goals of the pattern. Explicit manipulation of viewer model can be achieved by the use of these established dramatic patterns, for instance, surprise due to the unexpectedness of the rescuer in the deliverance example. Representing dramatic patterns as plan operators ensures that (i) actions are added to explicitly affect the viewer's perception of the story, and (ii) actions that threaten to break such explicit causal links are handled through the threat resolution process in the planning algorithm.

Current story generation systems focus on generating stories that are causally sound. That is, the stories' structure ensures that they execute correctly. While it is important to generate sound stories, there are certain actions and events in the story that are merely included for exposition or aesthetic value. Such actions may not appear in any important causal chain of the story leading up to the climax and are thus ignored by the algorithms. Representing dramatic patterns gives the author an opportunity to include actions that enhance the dramatic value of the stories but are not part of the causal chain of important events.

2.4 Film Theory Primer

Cinematography is the art of creating movies using a camera to film a story's unfolding action. It has developed into a successful and effective creative art-form in the past century. Film directors and cinematographers have developed effective visual storytelling techniques. They have also articulated various rules for using a camera to convey a story to a viewer. The stereotypical ways that a cinematographer uses a camera to film shot sequences are termed as Idioms. Cinematic idioms have developed from practical experiences as the film industry has progressed and some of them have been documented along the way. These idioms are basic rules for the composition of a shot and specify the way certain shots are to be filmed. (For example: How close should a Close-Up shot be ?). Cinematic rules also define concepts like the Line of Action (LOA) [1] or Thirds Grid [2]. Arijon [2] provides a comprehensive list of all the cinematic rules that are consistently used by cinematographers. We describe some essential cinematography concepts and techniques that are relevant to this work.

- **Composition**: Cinematographers tend to follow certain cinematic codes while organizing the visual elements within the frame of the camera. The general term for this set of codes is termed composition. For instance, the *rule of thirds* is a composition rule that can be obeyed by first imagining the frame to be divided into three vertical

---

[1] Imaginary line that passes through a character in the direction of action or movement

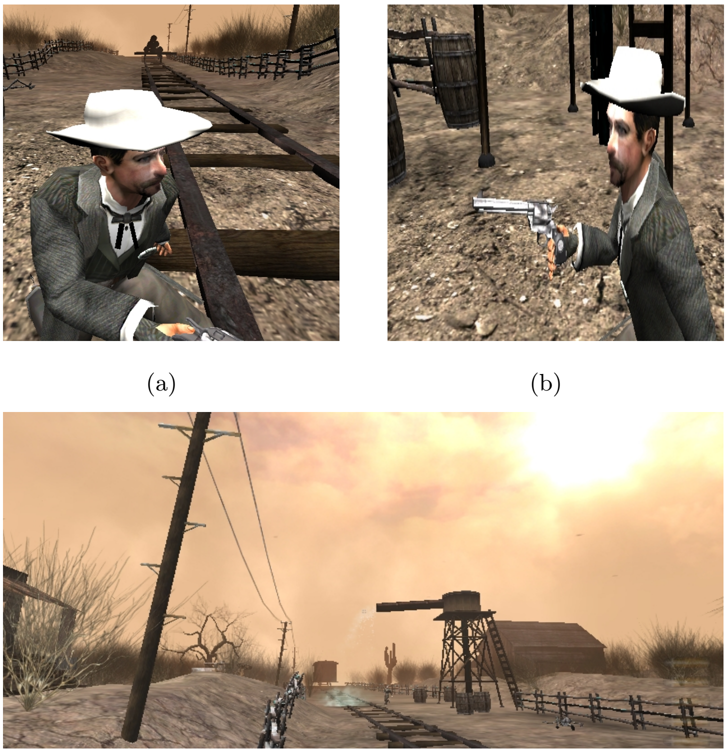
[2] Imaginary grid that divides the frame into three equal parts in horizontal and vertical directions

and horizontal parts, then framing each character or object so that its center aligns with the center of the thirds line. This is illustrated in figure 2.2(c).

- **Continuity**: Continuity in film refers to the consistent changes in camera placement such that viewers do not feel abrupt changes in their view of the story-world.. Continuity should be maintained temporally and spatially, both in the story world and the way it is presented to the viewer. Mascelli [3] provides a detailed description of various cinematic techniques for maintaining continuity. Continuity is essential for stories where the underlying context constantly changes over time and space. Cinematographers carefully select sequences of shots so that continuity is maintained. For instance, in Figure 2.1 the character's orientation cannot be ascertained by the viewer. Subfigure (c) shows an overview of the setting. From (a) it seems that the character is moving towards the house on the right of the screen. In (b) this is reversed and the character appears to be moving to the house on the right of the screen. This is misleading or confusing to the viewer and the continuity of shots is broken due to the sudden change in the camera's orientation.

- **Transitions**: Transitions are the visual punctuation of cinematic discourse. They are different techniques used to signal changes between consecutive camera shots. The most common transition is the *cut*, where one shot immediately follows another shot. Changes in story time can be depicted through a *dissolve* transition where consecutive shots overlap on the screen for a short period of time during the transition. A *fade out/in* transition is mainly used when the focus of the viewer is shifted to a different aspect of the story, where consecutive shots slowly fade in/out of a blank screen. Several other transition techniques have been documented in cinematography literature[3].

- **Coherence** : The camera is a communicative tool that conveys not just the occurrence of events, but also parameters like the mood of the scene, relationships that entities within the world have with other entities and the pace/tempo of the progression of the underlying narrative. In order for the narrative to be coherent, viewers must be able to identify these story parameters and understand the progression of events or development of character. For instance, in Figure 2.2, the telling of the narrative is enhanced by selection of camera angles such that the initial low angle



(a) (b)

(c)

Figure 2.1: Shots illustrating how change in camera position affects the viewer's orientation within the virtual world due to crossing line of action. In the first shot, the character appears to be moving from left to right relative to the viewer. In the second shot the character is moving from right of the screen to the left and it is not possible to know which direction the character is moving towards as there is no common static object in the frames. The layout of the scene is shown in (c). The character in shots (a) and (b) is standing near the water tank and is moving towards the house on the left of the screen.

(a) Dominant character expressed by using a low angle medium shot.

(b) Neutral character shown by a level close-up shot

(c) Character now suppressed as shown by a high angle shot

Figure 2.2: Shots illustrating how change in camera angle conveys the progression of a character in the narrative being dominant to submissive over time (from the movie *A Few Good Men*).

shot establishes dominance of Jack Nicholson's character who later turns submissive, indicated by a high angle shot from the frame of reference of the newly dominant character, Tom Cruise (seen here from behind).

2.5 Summary

- Overview of the different fields and techniques related to the problem of automatically generating movies.
- Survey of Geometric Camera Control systems and a discussion of their limitations.
- Survey of tools for supporting camera control and specific applications.
- Overview of Planning in Artificial Intelligence used for generating discourse.
- Brief overview of Dramatic Patterns from Narrative Theory and their computational formalization.
- Relevant cinematic techniques from Film Theory.

Chapter 3

A Planning Approach for Generating Cinematic Discourse

Story-based 3D virtual environments currently use fixed, pre-scripted camera shots to communicate information about the story to viewers. In dynamic virtual environments where storylines change subtly with each session of play, using pre-scripted shot sequences can limit a director's ability to express a wide range of communicative intentions tailored to the unique stories being told. The ability to tailor cinematics to context is needed, for example, in worlds where the physical configuration of the virtual environment is different each time the world is played. As another example, virtual environments such as educational games typically make use of user-specific pedagogical goals that differ on a per-session basis and create distinct narrative goals that need to be communicated effectively as the student plays through the game or training simulation.

In order to produce cinematics that are tailored to these varying contexts, a camera system is needed that can a) dynamically select shot sequences that satisfy novel communicative intentions and b) adapt to changing environments while maintaining cinematic quality. As discussed in the previous chapter, existing approaches using constraint-based algorithms specify positions for a virtual camera in response to geometric constraints in a dynamic 3D environments. These approaches handle physically changing environments, however, the knowledge they employ (in the form of constraints on camera positioning) constraints have to be hand-specified. Moreover, these approaches focus on static snapshots of the virtual world and do not address the ways that the camera can convey meaning that arises from the complex causal and temporal relationships between dynamic events within the environment.

My solution addresses these challenges by developing a camera system that performs three main tasks. First, it selects *salient* elements from the underlying context for communication. Second, it organizes the salient elements into a *coherent* discourse. Finally, it creates a visualization that follows cinematic conventions and schedules camera shots in a *temporally consistent* manner. A character, action or event of the story (collectively referred to as story *elements*) is *salient* to a narrative discourse if it is necessary to explicitly communicate some property of the element to the viewer in order to satisfy a communicative intention of the director. For characters, these properties typically involve aspects of the character's state while for actions and events these properties typically involve aspects of the element's dynamics (that is, that it occurred and how its occurrence has affected the world). A narrative discourse plan is *cinematically coherent* if it communicates the intentions of the director *correctly* and *effectively.*

Correctness means viewers are communicated the story as intended (which means that all communicative goals of the director are satisfied). Effectiveness of narrative communication, in this work, is judged by the ability of the viewer to identify the dramatic patterns that are used to convey the story. This is one restricted way of measuring effectiveness of narrative communication.

A visual narrative discourse plan is *temporally consistent* if the camera movement and transitions for viewing the story follow well-established temporal and cinematic constraints developed by cinematographers. These constraints, called *cinematic idioms*, relate common cinematic methods for filming actions to the various aspects of shots and shot sequences (e.g., shot angle, shot duration, shot movement). For example, a camera shot filming an action communicates different meaning about the action based on the duration of the shot. Because viewers implicitly understand these idioms and their use, it is essential for any schedule of camera shots to conform to these temporal constraints in order to correctly convey the specified communicative goals. Creating correct timing is complicated for systems that automatically generate shots, since the duration of camera shots is also dependent on the story actions they film; the execution times of these actions are often not known at design time. As a result, shot durations must be generated at run time to take both idiomatic and timing considerations into account.

3.1 Solution

In this work, I seek to meet the aforementioned challenges by providing a formalization of film idioms as communicative *plan operators*, a *plan generation algorithm* for constructing schedules for shot-sequences, and an execution environment running in a real-time game engine for visualization. The representation of camera control in terms of planning operators based on standard film language facilitates the automated generation of constraints (addressing Challenge 1) and provides higher-level reasoning mechanism for using established shot-organization idioms. The use of a planning formalism to create sequences of shots provides an explicit representational framework for story and camera actions, causal relationships, temporal relationships, and heuristics for specifying directorial preferences (addressing Challenge 2).

As mentioned above, camera shot selection and movement communicate information about a virtual environment and its story to viewers. Directors and cinematographers intentionally choose specific camera actions to manipulate viewer's beliefs about the storyworld. The process of constructing a movie can thus be seen, in part, as the task of selecting camera shots that achieve the goal of communicating a narrative. As discussed in Chapter 2, planning algorithms can be used to select operator sequences to achieve goal-directed behavior. More specifically, discourse planning algorithms that a) model communicative speech acts as plan operators and b) select sequences of communicative operators to achieve a communicative goal, have effectively been used to generate multi-sentential text-based communication. In this dissertation, I adapt approaches from discourse planning to the specific task of cinematic discourse generation.

The system I describe here, called *Darshak*, represents both primitive and more abstract camera operations as communicative operators that manipulate the beliefs of viewers. A hierarchical discourse planning algorithm is used to construct shot sequences that satisfy the communicative goals given to the system. Abstract operators represent narrative patterns and cinematic idioms, while primitive operators represent frames (or static camera shots) and camera movement specifications. Abstract operators affect the belief states of the viewer through focus shifts affected by the primitive operators. In order to relate the focus and temporal extent of a shot to the actions that it is intended to film, each camera operator contains temporal variables that are bound by the planning algorithm to actions in the story. During the plan construction process, the planner constructs a plan by instantiating actions that satisfy communicative goals or preconditions. Instantiated actions in the plan data structure are related to each other through causal links (added if the effect of one action satisfies the precondition of another action), ordering links (added to force execution of actions in a particular order), or decomposition links (added between abstract actions and the more-primitive actions that make up their sub-plans).

3.1.1 Characteristics of the Solution Revisited

The camera system developed in this work addresses the aforementioned problems by constructing narrative discourse from given information about story actions and their relationships. It selects *salient* actions and events from the story and associates them with appropriate communicative (camera) actions, organizes the story actions and camera actions in a *temporally consistent* manner, and ensures that the resulting narrative is *cinematically coherent*.

Saliency: In my approach, salient elements are chosen through binding constraints over story world knowledge imposed on the parameters of the communicative discourse operators. In order to effectively reason about communicating the story a director needs to take into account causal and temporal relationships between story actions. In my system, stories are represented as plan data structures - a specific representation proposed by Young [48] - with actions, causal links, and ordering links. A more complete description of story plans that serve as input to Darshak is provided in Section 3.2.1. Saliency is guaranteed due to the explicit representation of narrative patterns and cinematic idioms through binding constraints on the operators that constrain the binding of variables to salient elements in the story.

Coherence: Discourse plans generated by Darshak are coherent if they communicate the story correctly and effectively. In my work, correctness of a discourse plan is measured by determining the similarity between the story plan being communicated by the director and the story plan formed by the viewer after seeing the resulting cinematic. In this regard, correctness focuses on the correct *recognition* of the story structure by a viewer. Effectiveness is measured by determining the extent to which a viewer identifies the narrative patterns that are used by a director in constructing the narrative discourse. This property is closely related to the traditional notion of coherence in discourse as described in

Section 2.4. In this regard, effectiveness focuses on the correct recognition of the cinematic's discourse structure by the viewer.

Effective discourse in any medium typically follows a rhetorical structure in order to make the discourse actions coherent to the intended audience. In cinematic communication, a sequence of shots conveys coherent information if it is structured to satisfy certain rhetorical goals like elaboration or sequence. Decisions about the structure of a shot sequence essentially involve decisions about *content* (the determination of shot type as well as what things each shot films) and decisions about *organization* (the determination of the relative ordering of shots and their relationship to one another). These elements of cinematic discourse structure directly affect communicative phenomena such as suspense, for example, by the choices made to include or exclude certain objects or events in shots presented to the viewer. They signal the intended significance of elements from the story as well, often by the use of certain shot types or by the inclusion or exclusion of the story elements themselves. To represent the structural constraints of cinematic discourse, I use a hierarchical partial order causal link planning algorithm that has been successfully used to generate coherent natural language discourse [22] and argue that the same technique can be used to generate coherent cinematic discourse.

Temporal consistency: Time plays a central role in the telling of a story, both in terms of the timing of events within the story world as well as the timing of the communicative actions that are used in the story discourse to tell it. In cinematic discourse, these relationships are particularly complicated, as the camera action used to film a story action may start or end well before, well after or precisely at the corresponding begin or end time of the story event. Further, camera shots that communicate story actions also communicate the temporal relationships present in the content they are communicating. Events in the story that are temporally far apart in the story world but contiguous in the telling need to be explicitly separated by visual cues. Consider a scene of a young girl practicing her running skills in a field, followed immediately by a dissolve to the scene showing a grown woman running a marathon. Here the dissolve action clearly depicts the cue to the viewer that years have passed by in the story-world. In order to keep track of the temporal consistency of generated camera shots, my approach makes use of temporal indices included in the representation of cinematic plan operators, constructing temporal links between story and discourse plan elements to represent relationships between operators. A temporal consistency checking algorithm is introduced in the planner to evaluate temporal constraints on actions in a camera plan. This algorithm guarantees temporal consistency due to the explicit representation of temporal variables and the relationships between them specified in the operator definitions.

I argue that Darshak effectively addresses the two central challenges mentioned earlier. In Darshak, abstract plan operators encode dramatic patterns in discourse. Primitive plan operators and mid-level abstract plan operators represent cinematic camera placement idioms. The planning algorithm used by Darshak provides a mechanism for constructing cinematic discourse by reasoning about the causal and temporal relationships between the events happening in the story and communicative actions for filming the actions. Primitive camera actions establish geometric constraints that are identified and solved by a constraint solver for physical placement of the camera. Story actions are executed in the Zocalo [49] execution environment.

3.1.2 Components of the End-To-End System

The Darshak system is the central component of this work. Darshak is a hierarchical partial order causal link planner with temporal scheduling extensions. As shown in Figure 3.1, 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 actions, transitions, abstract cinematic idioms and narrative patterns. Camera placement and transition actions, represented as primitive operators, affect the focus of visual they represent 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. The story plan is input as a plan data structure that contains the description of the inital state of the story world, a set of goals, and a totally ordered sequence of actions and 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 are given to the system as a set of belief states for the viewer. The representation of a narrative and communicative operators for the camera and the declarative representation of the input story is described in Section 3.2.

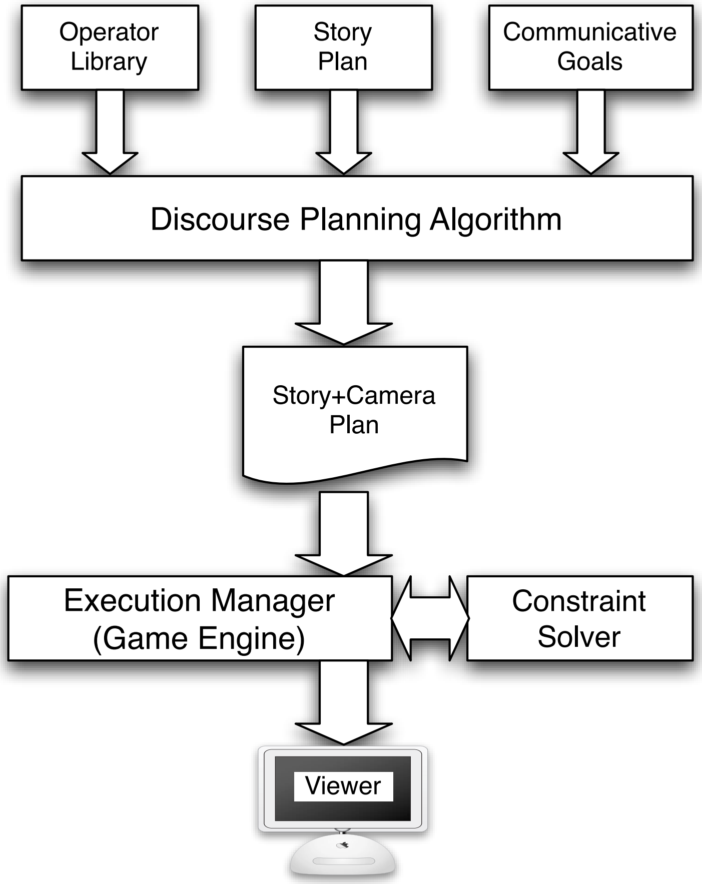


Figure 3.1: Darshak Architecture Overview

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, to the corresponding actions in the story plan. Details of the planner implementation are described in Section 3.3.

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. This camera plan is combined with the story plan and is written out by the system in an XML document. The XML document is sent to an execution manager running in the Unreal

Tournament game engine. The execution manager dispatches both story and camera actions by instantiating action classes on the game engine that represent code for performing the action in the engine. Story action classes on the game engine affect physical state of the 3D world, such as movement of characters. Camera actions impose viewing constraints on the game world's virtual camera. A constraint-solving algorithm constantly checks the viewing constraints and maintains the camera's parameters (location and orientation) such that the constraints set by the action classes are satisfied. The execution environment and procedural representation of story and camera actions are described in Section 3.5.

3.2 Representation of Narrative Discourse in a Planning Formalism

As a base model of narrative, I adopt in this work the bipartite representation of narrative proposed by narrative theorist Seymour Chatman [9] and others. In Chatman's 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 in the previous paragraphs operate at the discourse level. That is, they are elements of the communicative medium used to convey the underlying story content. Narrative discourse is dependent on the story for its content, drawn from the events, characters, settings and objects appearing in it. 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.

3.2.1 Declarative Story Representation

Story is one of the inputs to Darshak (Figure 3.1). The input story is in the form of a 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 processed and stored in the camera planner's knowledge base as domain information for the camera planning problem. Figure 3.1 shows the representation of a story that is input to the camera planning algorithm.

This representation provides a language for representing domain information that is used in communicative camera operators to refer to the story elements being filmed. In the

Table 3.1: 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”))
    (step s3) (act-type s3 refer) (agent s3 Royce) (secondary s3 Marguerite)
    (effect s3 (spoken Royce ”Good God Mama what is that thing?”))
    (effect s3 (refers Royce Marguerite object))
    (mood Royce angry (during s3))
```

following sections, I describe my formalization of the hierarchy of communicative operators used by Darshak across three layers, viz. Primitive Shots, Episodes/Abstract film idioms, and Dramatic Patterns. A library of communicative operators composed from these three layers is one input to the camera planning algorithm (Figure 3.1). These operators are defined the representation described in the following sections.

Darshak’s representation of operators allow operator-writers to define camera actions that ensure that *salient* elements of the story are communicated, multiple shots maintain *coherence*, and the resulting camera plans are temporally *consistent*. In the next few sections, I describe how this representation of camera shots as hierarchical plan operators captures these properties.

3.2.2 Discussion: Cinematic Language

Camera shots in a film communicate information about the story world to the viewer. Film theorists [2, 3] have described film as being composed of sequences of *frames* that collectively form *scenes*. A *Frame* is a snapshot of the story world at a any instant of time. A *scene* is a sequence of frames in which events in the story occur within the same spatial and temporal proximity. This is a non-traditional definition of the scene that I use in my work as it readily lends itself to a computational representation. In textual discourse there is a similar, though not identical, hierarchical structure where sentences collectively form paragraphs of text (Figure 3.2).

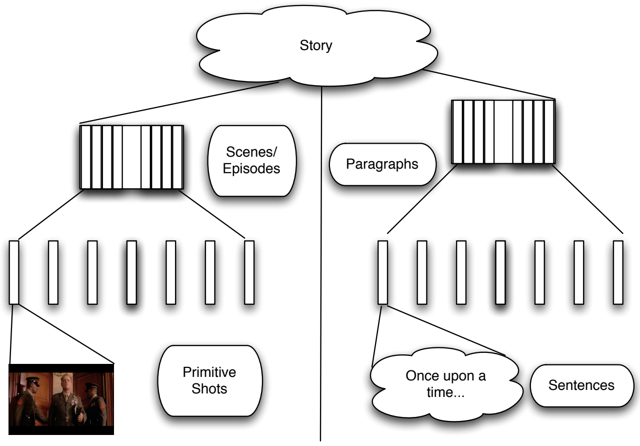


Figure 3.2: Hierarchical Representation of Narrative Structure in Film and Text. Camera shots are the surface/primitive actions, scenes are collection of spatially co-located and temporally contiguous primitive actions, and a collection of scenes together form the narrative.

Within a textual discourse, each paragraph of well-written text communicates a particular topic through careful placement and presentation of sentences [20] that individually and collectively contribute to the overall communicative intention of the paragraph. For instance, consider the following sentences: *Bob fired his gun at Jane. John fell on the ground.* These sentences are ambiguous because it is not clear how they are related. Adding one more sentence makes the discourse coherent and unambiguous. *Bob fired his gun at Jane. John fell on the ground. He had just jumped in the way of the bullet and sacrificed himself for his friend.* Thus, the information contained within each sentence, when taken individually does not communicate the information that is inferred when they are presented together in a discourse. As described in the Section 2, linguists describe text segments as collection of sentences that collectively convey communicative purpose. Each segment's purpose imposes constraints on individual sentences. For instance, according to one rule, anaphoric expressions are used under the constraints that they refer to the subject already introduced in previous sentences. Individually the sentence *He won.* communicates little information about the context in which a character 'won'. Now consider: *Michael fired an ace on match point. He won.* Here it is clear that the pronoun *He* refers to Michael because the anaphora can be resolved by looking at the relationship between the two sentences.

Just as in textual discourse, well-crafted films have discourse entities whose composition convey meaning. Films are composed of scenes that are, in turn, composed of shots that individually and collectively communicate the topic of the scene as intended by the director. Consider now a picture with a close shot of a man holding a trophy. This shot conveys that the man is holding a trophy and does not tell us anything about the context. If the same shot is followed by a wide shot of a tennis stadium with applauding audience then the two shots put together provide additional information about the context. It can be seen from this example that there are parallels between inter-sentence relationships seen in natural language and inter-shot relationships in the visual medium.

3.2.3 Primitive Shots

A *Primitive Shot* in Darshak is defined as an entity that describes a set of parameters for the physical camera. For instance the shot in Figure 3.3 is a close up of a character shown from the front. The operator that describes the shot is shown in Table 3.2.

Table 3.2: LookAt Operator

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

Within the literature on film, it is clear that cinematographers typically make use of only a relatively small number of primitive camera shots[2]. Primitive shots 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 focus shifts in a coherent manner scenes are built that communicate the story. Primitive

Figure 3.3: Shot Corresponding to Operator LookAt

shots as plan operators capture certain coherence and shot-composition rules. For example, in order to avoid disorienting the viewer, one of the preconditions of a tracking shot is that the actor or object that the operator is tracking should be in focus before movement starts. The preconditions (infocus ?f)@[Tstart) ensures that a jump to tracking camera is disallowed if there is 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. Shots like the Wide-Angle shot with a balanced composition and continuous motion can be used as as establishing shots. Shot composition can also be used to depict shift in fortunes when a stronger character emerges and challenges evil antagonist successfully. Jennifer Van Sijll, in her book *Cinematic Storytelling* [50], 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. My representation of operators captures a subset of these dramatic effects of camera actions. In my work, the dramatic value is determined by the higher level communicative pattern (that is, the abstract plan operator) that includes the shot. I use importance metrics of actions in the story to determine the dramatic value. These metrics inform the selection of shots that communicate the desired dramatic effects. The selection of a shot with the appropriate dramatic value is carried out through propagation of constraints along the hierarchical operators. Abstract operators are described in the next section. I have implemented 9 primitive operators and several variations of these operators. 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. A complete list of all operators defined for use in Darshak is provided in Appendix A.

Table 3.3: Track Operator

Operator
Type : Track
Parameters : ?f, ?shot-type, ?dir
Preconditions :
(infocus ?f)@[Tstart)
(not (tracking ?f ?shot-type (rev ?dir))@[T1, Tstart))
Constraints : (>Tend Tstart)
Effects :
(infocus ?f)@[Tstart, Tend)
(tracking ?f ?shot-type ?dir)@[Tstart, Tend)

3.2.4 Episodes/Abstract Scenes

Film idioms lend themselves to a hierarchical representation, with sub-parts 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 to the mental states of the viewer. They also encode relationships between primitive operators and provide a mechanism for expressing established idioms as specific decompositions. Individual shots like the ones described in the previous section have a denotative meaning associated with them that focus the viewer's attention to elements in the frame. There are relationships between shots that convey more than the information that is present in the individual shots in isolation.

Abstract plan operators capture such abstract communicative phenomena by explicitly representing the different ways in which such information can be conveyed. Further, additional elements in a narrative can be captured through plan operators. For instance, consider the example of conveying the personality of the character in a story using specific shot types and shot sequences. Figure 2.2(a) shows how a character starts out as being viewed as dominant and then progressively becomes portrayed as submissive as the story goes on.

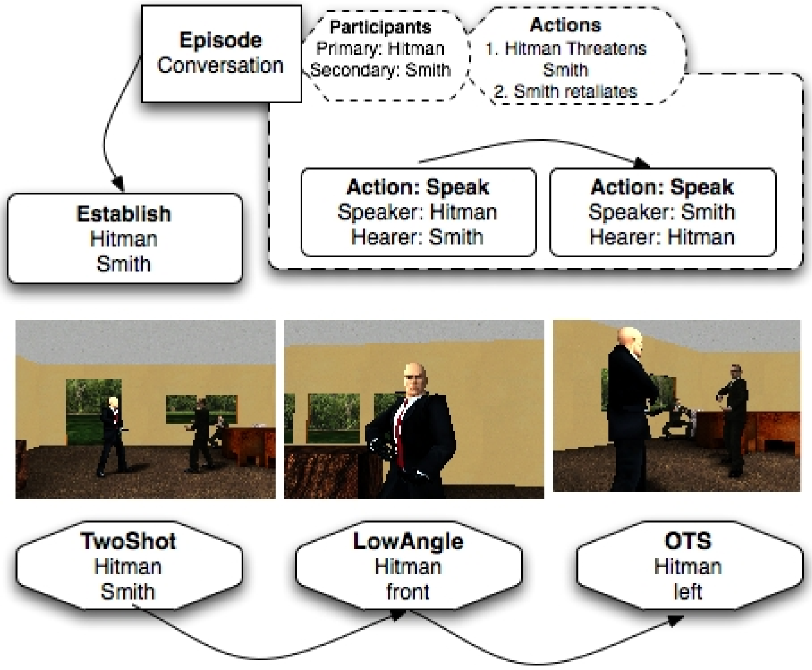


Figure 3.4: Illustration of the conversation episode and it's constituent primitive actions. Here squared box represents an expanded abstract action, dotted bubbles next to the boxes indicate the parameters of the action, rounded boxes represent unexpanded and primitive actions, hexagonal boxes represent camera actions. Arrows indicate temporal relationships between actions

Consider the example shown in Figure 3.4. Here the scene involves a conversation between two characters: Hitman and Smith. Hitman is the primary character and in the constituent action is threatening Smith. In response to this threat, Smith is retaliating with an aggressive gesture. The episode is coherent to the viewer if the location and characters are familiar. This is represented by preconditions on the operator viz. (establish ?doc), (establish ?primary), and (establish ?secondary). The sub-goals of this conversation are (infocus ?primary)@[t_{se}, t_{ee}) and (infocus ?secondary)@[t_{sa2}, t_{ea2}), that the primary character should be in focus throughout the conversation and the secondary character should be in focus during the second speech act.

Importance of abstract operators

Typically, authors have some form of high-level communicative goals in mind when they set about to create narratives. In film, for example, there are some goals that the director works to establish and maintain throughout the unfolding of the plot. These high level goals are pragmatic in nature and are not satisfied by just the execution of one single communicative action. These goals are similar to the stylistic goals, sch as formality, in text generation systems such as Pauline [45]. The establishment of these types of goals is addressed in Darshak by the use of abstract actions and hierarchical decompositions. Such goals act as pragmatic constraints on the selection of operators during the planning process. Narrative tempo and character emotion are two examples of these types of constraints. While Darshak does not have detailed emotion or tempo models, one can imagine extensions to Darshak's representation that made use of constraints on the use of particular decomposition operators to create narratives with particular types of tempo or the communication of a certain progression of character emotion.

Here I show with a simple example how a plan operator representation captures the process of encoding the cinematic idioms as decompositions on an abstract operator with an emphasis on specific cinematic properties such as tempo and emotion.

Maintaining Tempo High tempo is characterized by an increased number of transitions (e.g., cuts) and motion shots (e.g., pans, tilts). Lower tempo results in selection of shots with longer duration. In the example operator shown in Table 3.4, a low tempo value will lead to the refinement of the abstract conversation operator into a decomposition that contains only a 2-shot lasting the whole duration of the conversation. The start and end time variables of the operator are temporally constrained to match with the times of the constituent actions in the conversation.

Table 3.4: Example of incorporating tempo in operator representation

Operator: Show-Conversation @ [tstart, tend)
Parameters: ?c1 ?act-list, ?tstart, ?tend
Preconditions:
Constraints: (conv-step ?c1 ?a1) (conv-step ?c1 ?a2)
Temporal Constraints: (= ?tstart ?ta1start)
Effects: (BEL V (Occurred ?c1)@[?tstart, ?tend))

Decomposition: Show-Conversation
Parameters: ?act-list, ?tstart, ?tend
Preconditions:
Constraints: (tempo low)@[?tstart, ?tend)
Temporal Constraints: (after step1 step2)
Steps: (step1 Master-Shot ?c1)
(step2 Apex-Shot ?c1)
Effects: (BEL V (Occurred ?conv)@[?tstart, ?tend))

Depicting Emotion One of the ways of depicting heightened emotion is through high-angle (negative valence) or low-angle (positive valence) shots. This can be represented with the plan operators shown in Table 3.5.

Automated text-based discourse generation programs [44] have used this abstract operator formalism to encode rhetorical relationships between text segments. Rhetorical relationships similar to those found in text also exist between camera shots. Since the representation that we use is similar to that used in traditional text-based systems, similar rhetorical operators can be used for maintaining rhetorical coherence across multiple shots. For instance, the rhetorical strategy of elaboration in visuals could be represented by first having an overview shot followed by close shots focusing on specific details of the scene. Thus hierarchical discourse plan operators are also imporant to structure visual discourse, although the rhetorical conventions of shots have been less formally documented.

3.2.5 Dramatic Patterns in Narrative

Focus is one of the important aspects of effective storytelling. In this dissertation, focus refers not to the notion of the clarity of a camera shot, but rather to the property

Table 3.5: Example of incorporating emotion in operator representation

Operator: Show-Dialog @ [tstart, tend)
Parameters: ?s, ?tstart, ?tend
Preconditions:
Constraints: (speak ?s) (actor ?a ?s)
Temporal Constraints: (= ?tstart ?ts1start)
Effects: (BEL V (Occurred ?s1)@[?tstart, ?tend))

Decomposition: Show-Dialog @[tstart, tend)
Parameters: ?act-list, ?tstart, ?tend
Preconditions:
Constraints: (emotion high)@[t1, t2) (emotion low@[t2, t3)
Temporal Constraints: (after step1 step2) (¡ t1 t2) (¡ t2 t3)
Steps: (step1 (Look-At-Close-Low ?a1 t1, t2))
(step2 (Zoom-Out-Low-Med ?a1, t2, t3))
Effects: (BEL V (Occurred ?s1)@[?tstart, ?tend))

of cinematic discourse that ensures that all *salient* elements of the story are attended to in the minds of viewers during the telling of the narrative. Selection of salient elements is motivated by the events happening in the underlying 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. I operationalize these narrative patterns as plan operators. This representation allows the planning algorithm to select salient actions from the story for presentation.

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 Supplication defined in the following paragraph there are three participants each with characteristics like sinner, punisher etc. These participants are 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 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.

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 decomposition operators, all decomposing the same abstract action but each with varying sets of applicability constraints.

The set of dramatic patterns that I draw upon were identified by Polti based on exhaustive analysis of a large corpus of popular stories [46], ranging in genre from Greek mythology to late 19th century literature. These patterns were initially identified and individually described in much detail but subsequent to their definition there has not been an attempt to describe the relationships *between* these patterns. In this work, I extend the patterns identified by Polti by adding 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 between abstract operators.

To better illustrate the formalization of dramatic patterns, consider the operator representation of the pattern *Deliverance* shown below. Deliverance, as described by Polti, occurs when an unfortunate individual or group that are commanded by a threatener get 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 (i) there is a threatener who is controlling the unfortunate (ii) the unfortunate do not have hope of escaping and (iii) 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 on 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 to expect the rescuer to fail by showing examples of his or her earlier failed attempts. The operator representation of Deliverance is shown (figure 3.6).

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)

Table 3.6: Operator definition of *deliverance* pattern

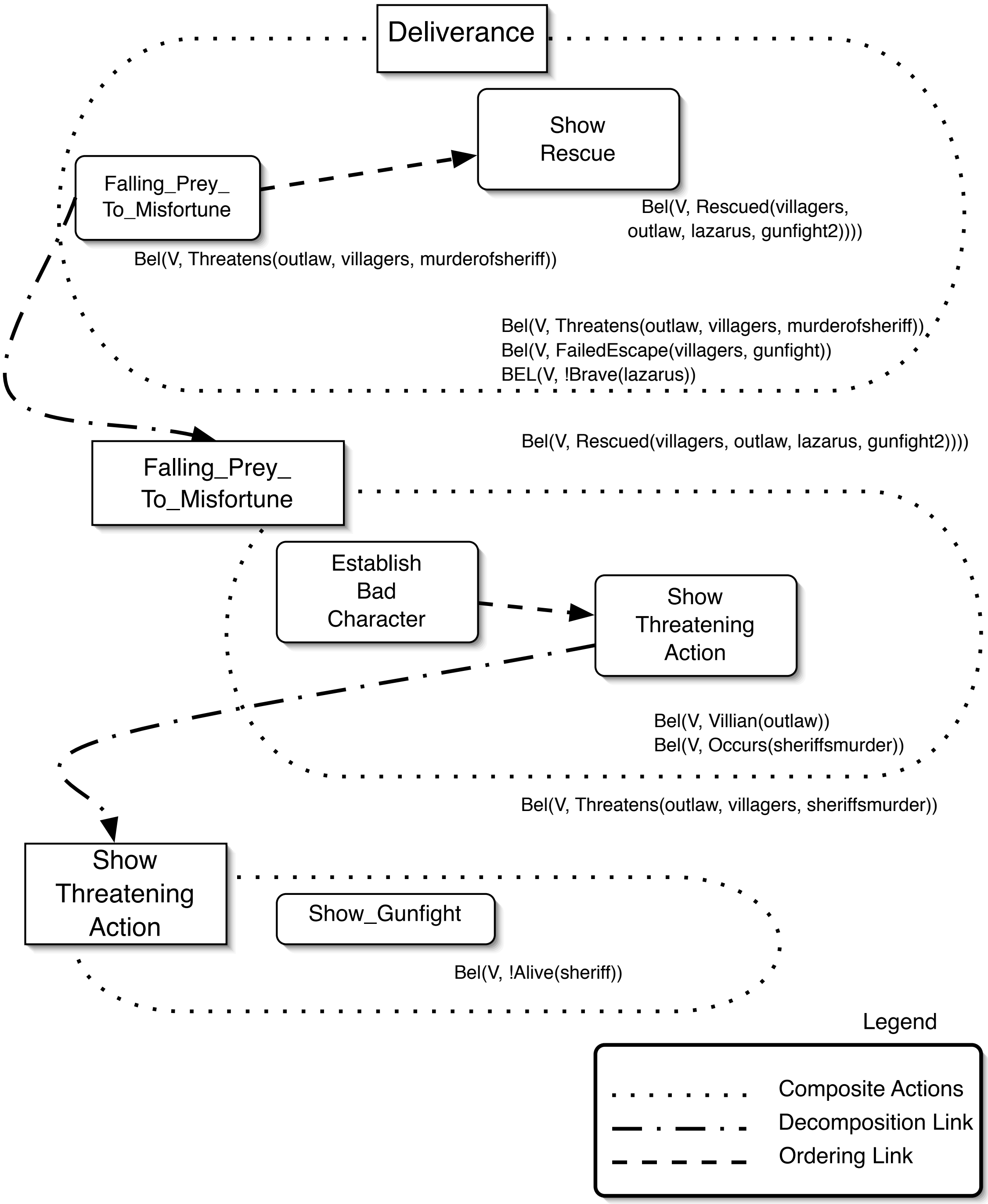


Figure 3.5: Part of the story that illustrates the relationships between narrative patterns. *Deliverance* pattern is used when an unlikely hero saves the unfortunate from the hands of an evil power. In a story told using this pattern the author establishes the evil character by showing some action that leads to the evil character overpowering the weak (gunfight to kill the sheriff in this example). The author then tells the actions that lead to the reversal of fortune (rescue of the weak).

3.3 DPOCL-T Algorithm for Generating Cinematic Discourse

Planning algorithms have been successfully used in the generation of effective textual discourse [44] as well as for story generation [51]. In this work, I adopt the model of narrative plot described by several narrative theorists such as [10, 52, 53, 46, 54, 55]. As described in the previous section, my representation of narrative plans contains 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. The planning algorithm is a program that makes use of the story representation, described in the previous section, to create and modify narrative-structured plans and organizes the narrative plans to structure discourse actions. The program itself has two parts: The AI planning algorithm described in detail in this section, and an execution-monitoring component running within a game engine that is described in the following section.

The representation of the actions in novel, dynamically created stories is facilitated by the use of a planning formalism, as this approach is capabable of generating plans that are structured to exploit both the parameters of actions as well as the causal/temporal relationships between them. Traditional planning algorithms (like DPOCL) involve two kinds of reasoning. First, they select the actions that achieve or help achieve the goals of the planning problem. For example, in order to achieve the goal of filling up a bottle of water, the planner might decide to perform actions of putting the bottle under a fountain and filling it up. Second, for the case of hierarchical planners, they involve selecting the appropriate set of primitive actions from multiple ways of achieving the goals of higher-level actions. For example, filling up a bottle of water under a fountain may involve putting the bottle under the fountain, pressing the button to turn the fountain on and then releasing the button after the bottle is filled.

This work demonstrates that classical planning formalisms can be effective for representing narrative discourse. However, the approach has several disadvantages that need to be overcome for providing a rich representation of all the essential elements of narrative discourse. One assumption made in traditional planning is that actions execute instantaneously and assert their effects immediately. Another assumption that is often made is that any effect asserted by an action hold unless another action explicitly changes it. While careful operator design can circumvent these problems, often this careful design makes operator definitions unintuitive. To address these issues in a more clear way, Darshak's operator representation is annotated with temporal variables that are related to and constrained by events happening in the story world. The details of the temporal annotations and extensions to the planning algorithm are described in detail in this section.

3.3.1 Extensions to the Planning Algorithm in Darshak

The elements that are included in the representation of DPOCL-T planning domains and plans are the following:

- *Constant symbols* describe the objects in the domain.
- *Variable symbols* are of two types viz. object variables that range over the set of all constant symbols, and temporal variables that are constrained through temporal relations.
- *Object Binding Constraints* limit the possible values for object variables from the domain.
- *Temporal Constraints* are maintained between temporal variables and are required to be consistent through the execution of a plan. Temporal constraints are restricted to two types: $<$ and $=$. This enables us to represent the time variables along a graph with directed edges representing constraints.
- *Rigid Conditions* are conditions that cannot change over time. For instance, the connectedness property of locations can be expressed as *adjacent(loc1, loc2)* and cannot change.
- *Flexible Conditions* are conditions that can change over time through the effects established by actions executing in the world. For Instance, *open(door1)* may change if the door is closed as a result of the execution of some action.
- *Temporally Indexed Conditions* are of the form

$$P(x1, x2, x3, ...)@[t_s, t_e) \tag{3.1}$$

where P is a flexible condition that holds for any time t such that $t_s \leq$ t $<t_e$.

3.3.2 Definitions

DPOCL-T supports durative actions with temporal constraints on temporal variables. Actions are defined through action schemata as well as a possibly empty set of decomposition schemata. These are defined as follows:

Definition 9 *(***Action Schemata***). An action schemata is a tuple $\langle A, V, \tau, P, E, B, \Gamma\rangle$ where A is the action type, V is a set of free object variables, τ is a set of temporal variables, P is a set of temporally indexed preconditions of the action, E is a set of temporally indexed effects of the action, B is a set of binding constraints on the variables in V and Γ is a set of temporal constraints on the variables in τ.*

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 τ implicitly contains t_s and t_e variables that denote the start and end of the instantiated action.

Actions that are *primitive* can be directly executed in the world and contain an empty decomposition schemata. *Abstract* actions have one or more decomposition schemata associated with them as recipes for decomposition.

Definition 10 *(***Decomposition Schemata***) A decomposition schemata is a tuple $\langle$A, S, B, O, L$\rangle$ where A is the action type, S is a set of step identifiers, B is a set of binding constraints on free variables in S, O is a set of temporal constraints on the temporal variables in S, and L is the set of causal links between the steps in S.*

3.3.3 Domain, Problem and Plan

The domain of DPOCL-T comprises temporal variables in addition to object variables, constants and operator definitions. Formally,

Definition 11 *(***DPOCL-T Domain***) DPOCL-T domain is defined as the tuple D=$\langle$C, V, τ, L, O$\rangle$ where C is the set of constant symbols, V is a set of variable symbols, τ is the set of time variables, L is the set of conditions (rigid and flexible), and O is the set of operators.*

Since the conditions are temporally indexed the initial state of the problem not only specifies the state of the world at the beginning of execution but also the actions and events that occur in the future that are not caused by execution of actions in the plan. This feature improves the expressiveness as temporal information about the story world actions that occur in a narrative can be explicitly conveyed to the camera planner though the initial state input.

Definition 12 *(***DPOCL-T Problem***) A DPOCL-T problem is defined as a tuple* $\langle D, \Phi i, \Phi g \rangle$ *where* D *is the domain of the problem,* Φi *and* Φg *are the initial and goal states of the plan respectively.*

The plan definition in DPOCL-T is similar to DPOCL except the manner in which ordering is specified. In DPOCL the partial ordering is expressed through ordering links. In DPOCL-T the ordering is implicitly expressed by the temporally constrained variables of the steps.

Definition 13 *(***DPOCL-T Plan***) A DPOCL-T plan is a tuple* $\langle S, B, \tau, L_C, L_D \rangle$ *where* S *is a set of steps,* B *is a set of binding constraints on free variables in* S*,* τ *is the set of temporal constraints on time variables in* S*,* L_C *is the set of causal links between steps in* S *and* L_D *is the set of decomposition links among steps in* S*.*

3.3.4 DPOCL-T Algorithm

Given a problem definition and the domain description described in the previous section, the planning algorithm generates a space of possible plans that satisfy the goals specified in the problem starting from the initial state. The DPOCL-T algorithm generates plans through a refinement search through this space.

Causal Planning

In conventional planning algorithms, an action is added to the plan when one of its effects establishes a precondition of another step in the plan. A causal link is added to the plan that connects the two causally connected steps. In DPOCL-T causal planning also requires enforcing temporal constraints on the intervals in which the precondition and effects hold.

Definition 14 *(***Causal Link***) A causal link is defined as* $L_C = \langle S_i, S_j, C\rangle$ *where* S_i *is the step whose effect establishes a precondition C of step* S_j*.*

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.

Figure 3.6: Sketch of the Camera Planning Algorithm

If an action A has a precondition P_i that is supposed to hold at time t for enabling A's execution and another step B has an effect P_i that holds over the interval $[t_1, t_2)$ then the establishment of a causal link between steps A and B will lead to an update on the constraint list with the constraint $t_1 \leq t < t_2$. Adopting the term used by [56], I will call the interval $[t_1$, t) and the constraints associated with the interval as a *Protection Interval* on the condition P_i.

Decompositional Planning

Decompositions in DPOCL-T are similar to their counterparts in DPOCL, except that they differ in the representation of ordering links between step stubs in the decomposition. Since the action specification explicitly includes time intervals, partial ordering is specified through constraints on the time variables involved in the decomposition.

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. In DPOCL-T, a global list of constraints is maintained as a directed graph of all the time variables involved in the plan steps. Whenever the planner adds a step, all the time variables are added to the directed graph and the graph is completed with the constraints representing the edges of the graph. A condition is threatened by a step if the effect of the step changes the condition within its protection interval. Such threats can be detected by finding a cycle in the temporal constraints graph. The computational overhead for finding a cycle in the graph is less if the graph is maintained progressively as steps are added. It can be observed that if a temporally consistent graph with no cycles contains a cycle on addition of a node then that node is included in the cycle.

3.4 Generation of Cinematic Discourse in Darshak

The main objective of this work is the generation and execution of coherent cinematic discourse. At the beginning of this chapter, I argued that a rich representation of cinematic idioms and narrative patterns is needed to generate coherent discourse. One of the challenges of developing a computational model of visual discourse is the creation of a mapping from notions from cinematography and directing into a representation that is favorable to computational manipulation. In section 3.2, I presented a systematic formalization of cinematic conventions and narrative patterns into a plan operator language. In that section, the semantics of various elements of a plan operator are described with respect to the cinematic language. Primitive operators represent camera shots that can execute to impose geometric constraints on the virtual camera. The library of primitive operator consists of well-known and commonly used shot compositions. Abstract discourse operators contain links to actions in the story and bindings to objects in the story, and constrain the primitive actions to focus on only the salient elements of the story.

As discussed in Chapter 1, maintaining viewers' focus on salient elements of the story is important for generating coherent discourse. Darshak takes advantage of a viewer's familiarity with narrative patterns, and those patterns' visual depiction through idioms to constrain the large number of possible camera viewpoints into a small set of specific shot compositions influenced by the context of the story. In addition to constraining the search space of possible shots, I argue that identification of narrative patterns leads to better comprehension in viewers. While it is harder to measure the degree of pattern identification among viewers, it is possible to measure elements of viewers' comprehension of the story and make relative comparisons between comprehension of the same story when visual discourse is generated both with and without intentionally choosing narrative patterns to influence selection of shot idioms. In Chapter 4 I present results from a user study based on established cognitive models of story understanding.

Visual discourse actions are durative in nature. There are specific temporal relationships between discourse actions and story actions. Unlike textual communicative actions that describe story actions, the performance of visual discourse actions and the execution of story actions in a virtual world need to be precisely timed and coordinated. A slight delay in camera movement significantly affects a viewer's orientation and distracts him or her from the context of the story. Maintaining temporal consistency of camera movements and story-world actions is another challenge for systems generating effective visual discourse.

Traditional discourse planning algorithms do not have explicit representations of temporal properties of durative operators. Also, they do not contain explicit links to actions in the story world. In textual discourse, story context is embedded within the predicates describing the conditions of the story. In Section 3.3 I describe the DPOCL-T algorithm that extends an existing textual discourse planning algorithm to tackle these unique challenges of visual discourse. An explicit representation of temporal variables and relationships ensures that discoure resulting from DPOCL-T's reasoning algorithms is executed in a temporally consistent manner.

The details of the planning algorithm's use of cinematic discourse operators to generate camera plans is illustrated in the next section through a worked example.

3.4.1 An Example

The DPOCL-T algorithm is illustrated by the following story about a thief, Lazarus Lane, who goes to a town in Lincoln County, Nevada 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 3.7. The example traces a single path through the discourse plan search space that DPOCL-T generated. The initial state contains sentences about the story actions. The goal state, in this example, 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.

Table 3.7: 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 deputy sheriff
Step 4 Lane overhears that Sheriff Bob has cancelled the plans of 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

As an open condition for the planner, the goal sentence 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 at an abstract level 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. The

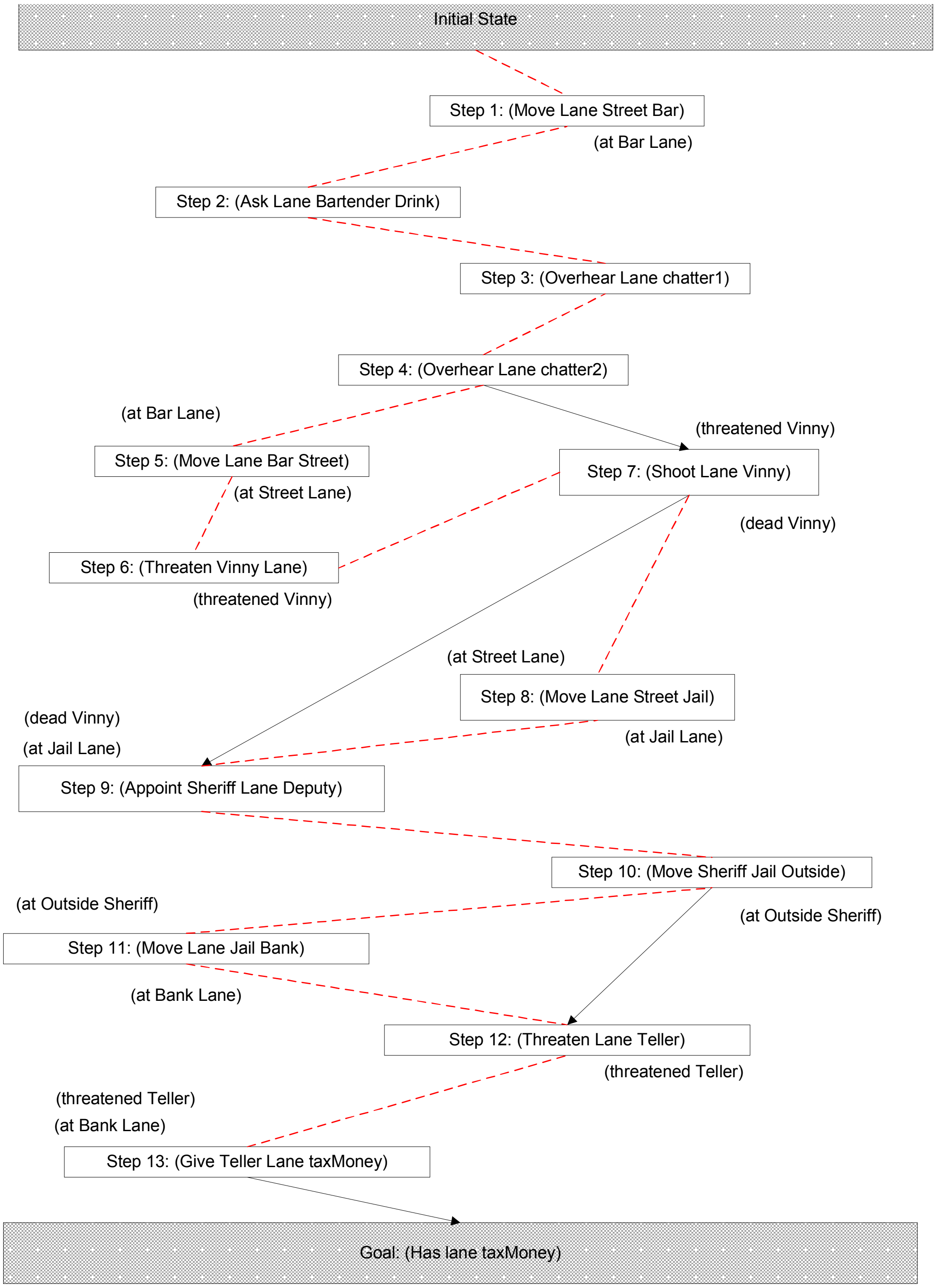


Figure 3.7: Simplified pictorial representation of the example story plan input to the planner. Boxes represent story actions, red dotted lines indicate ordering links, and solid black arrows denote causal links

planner could have chosen any of the other patterns with the effect (Bel V (Has ?character ?object)) that would have satisfied the story world's constraints. Given this choice, the planner instantiates the action and adds it to the discourse plan, and updates the open conditions list with the pre-conditions of the Show-Robbery action.

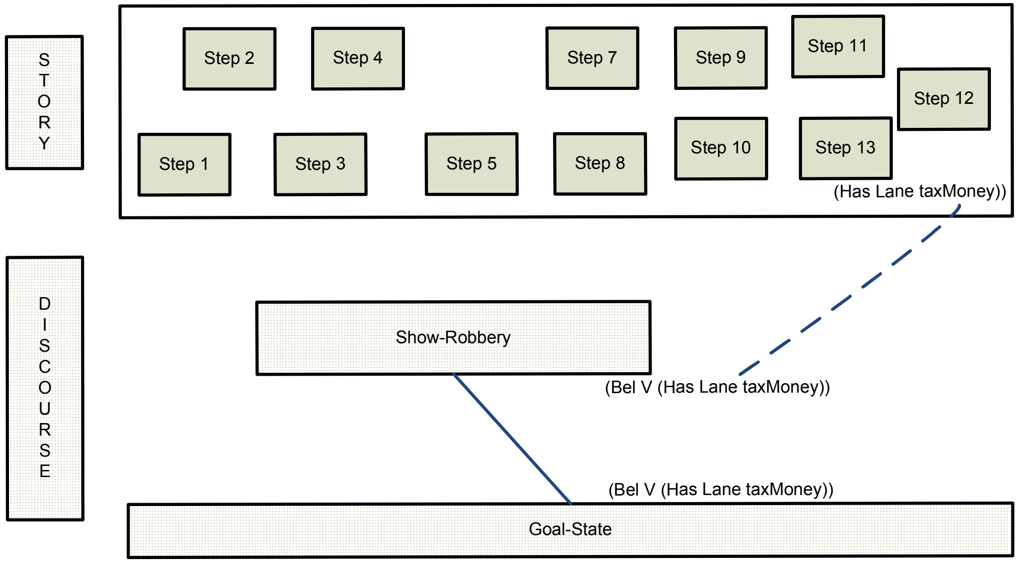


Figure 3.8: Initial step in the discourse planning process with a single unexpanded step that serves the lone goal of the problem.

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, Show-Resolution. The discourse actions are bound to story actions through 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 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.

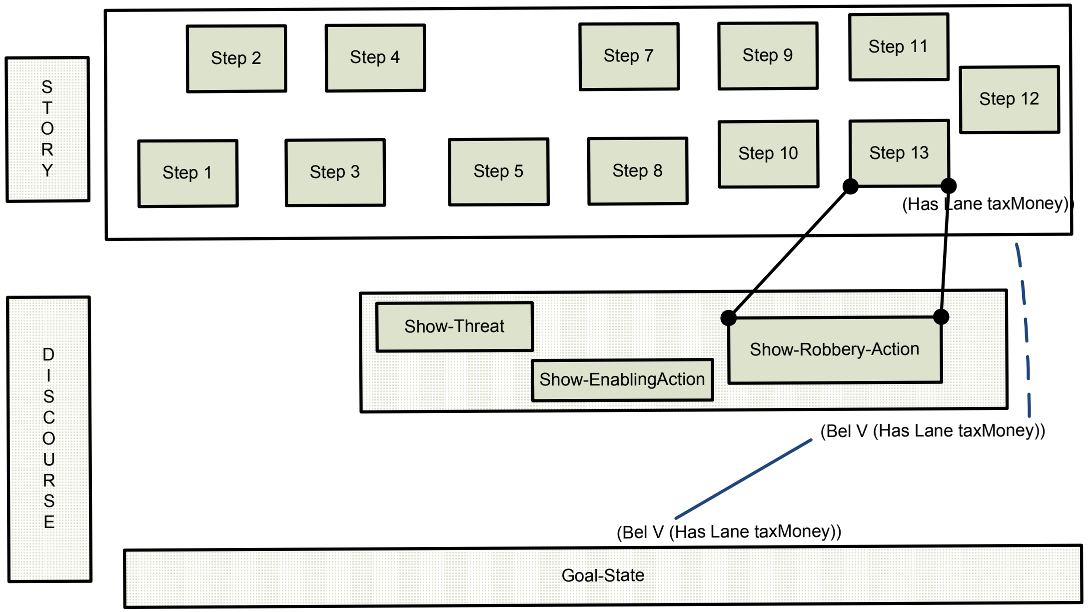


Figure 3.9: Second step in the planning process, the unexpanded step is now expanded and variables are bound.

The planning process continues by expanding the Show-Robbery-Action further and chooses a Medium-Two-Shot primitive action with primary character bound to Lane and secondary character bound to teller Elsie. This is done through constraints on the Medium-Two-Shot operator that query the knowledge base for (agent Step13 Lane) and bind the primary and secondary characters for the camera shot. The start and end time variables for the Medium-Two-Shot are bound to the start and end of the step's parent action, which, in turn are bound to the start and end times for the story action. The Medium-Two-Shot primitive action with the bindings for primary and secondary characters and time variables is then added to one of the branches in the plan space.

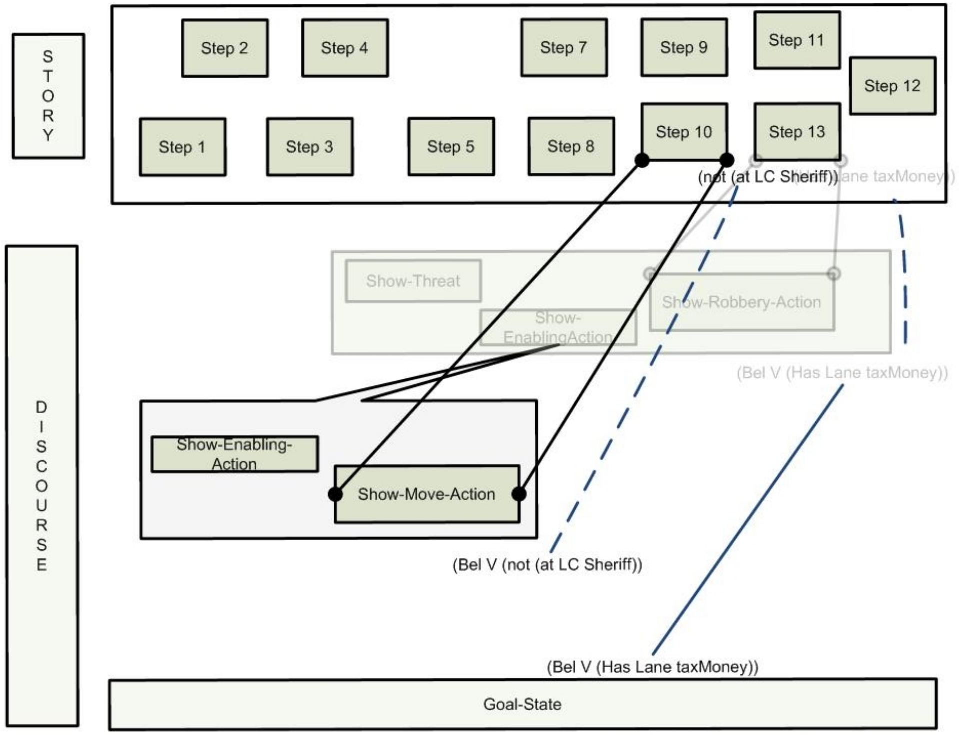


Figure 3.10: Another step in the planning process, the unexpanded step is now expanded and variables are bound.

During execution in the virtual environment, the Medium-Two-Shot action class defined on the game engine adds geometric constraints for composing the two-shot for filming Step13 in the story at the time of its execution. The planner maintains a simple temporal graph (STG) [57] of all the 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.

3.5 System Architecture and Execution Environment

The execution management for Darshak's output plans on the game engine is carried out within the Zocalo framework. The components of the end-to-end system are shown in Figure 3.12. 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 partial order causal link planner - Darshak - through a web-service called Fletcher [58]. The execution of planned stories and camera sequences occurs on a game engine (Unreal Tournament 2004) through an execution manager that communicates with the web-service. Several tools are available to the user for specifying domain information and providing heuristics to guide the planner's search algorithm [59]. The following sections describe the components of the Zocalo architecture and their role in generation and execution of dynamic stories as well as cinematic discourse for visuallizing these stories.

3.5.1 Zocalo: A Service Oriented Architecture

Zocalo is a service-oriented architecture for intelligent control of procedural elements within a game engine for the purpose of interactive story and cinematic discourse generation[58]. Zocalo provides general support for connecting various intelligent algorithms to a wide variety of execution environments (e.g., game engines, virtual worlds) through a well-defined service-oriented interface. The use of the Zocalo architecture eases two challenges faced by developers of interactive AI applications: First, it offloads the heavy computing resource requirements of partial-order planning algorithms (such as the

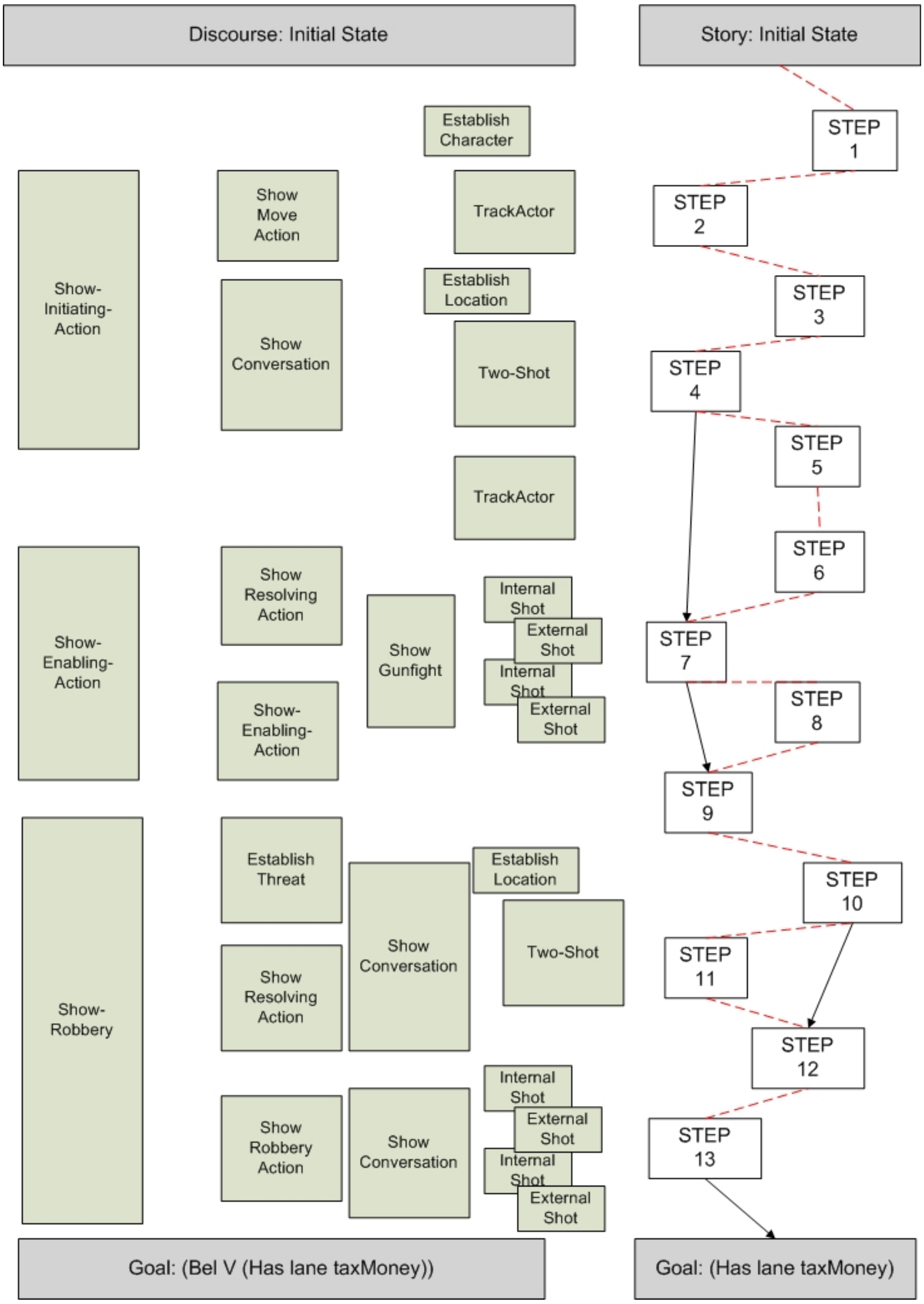


Figure 3.11: Fully realized discourse plan. The story plan steps are shown on the right. The discourse plan actions are shown hierarchically from left to right. Right most actions are primitive actions and actions to the left are abstract actions. Primitive camera actions film the story world actions 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.

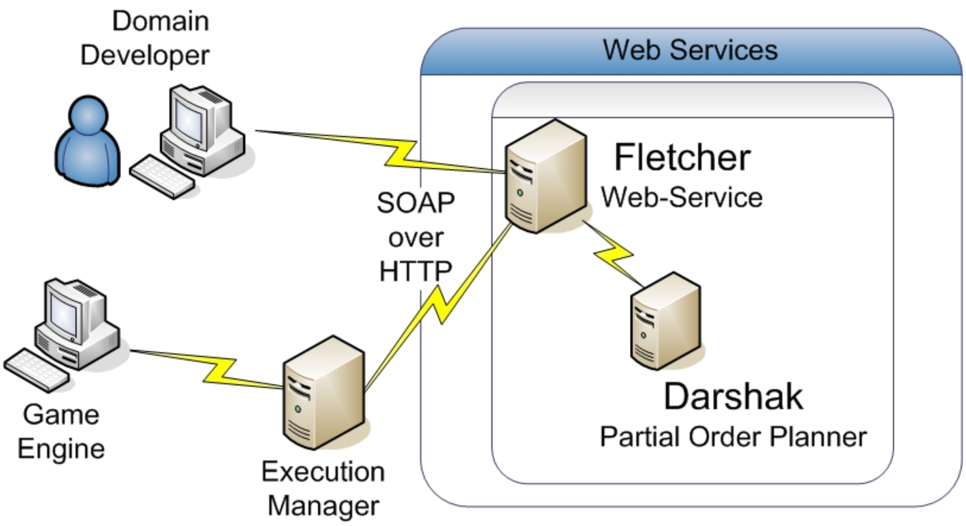


Figure 3.12: Zocalo Architecture overview

ones used in my work) or other intelligent processes to a remote server, freeing computational resources on client machines for use with the computationally expensive game engine algorithms. Second,it provides a two-way communication layer between the game engine and the planning algorithms so that user's actions occurring in isolation on the game engine are communicated back to the planner for reconciliation with the causally coherent story plan.

The Zocalo architecture has four central elements:

- An application library containing XML schema definitions and data utilities for translating information across different applications requesting and providing services through Zocalo.
- Services that compute story and discourse plans and perform additional analysis on the stories.
- An execution manager that serves as an intermediate module between the planning services and the game engine environments. Each execution environment has its own implementation of the execution manager that contains information about the specific code to manage instantiation, execution, and monitoring of actions within that particular environment.

- The execution environment receives action directives from th execution manager and translates these directives into specific function calls in the game engine.

In this work, I have chosen the Unreal Tournament game engine as the execution environment and utilized the existing interface to Zocalo services for character control within the engine. I have extended the execution environment to control the virtual camera in the game with directives from the planning service.

Execution Management on the Game Engine

The Unreal Tournament game engine is extended with a small interface that allows the execution manager to send instructions for controlling action execution within the engine. When the engine receives an instruction for executing an action, the game finds the functions responsible for performing the particular action that was requested. For this, the game engine first instantiates an action class that encodes the behavior corresponding to the execution of that action. Next, it adds the action to the list of actions that are ready to execute. The list of ready actions is continuously checked by a module called the *Action Executor*, which initiates the execution of all the actions that have their pre-conditions and temporal execution constraints satisfied in the game world.

Action classes are functional definitions (i.e. code written in the native environment of the game engine) corresponding to the plan operators that are declaratively represented in the planner. An example of an action class is shown in Figure 3.13. Each action class contains a list of variables that correspond to the variables in the plan operator, described in the previous chapter. These variables are bound to the identifiers of specific objects in the game world. For instance, in Figure 3.13, the Shoot action contains a variable for the weapon that is used for the shooting action. This variable is defined to be of type Weapon, which is the superclass of all weapons in the Unreal code library. When this class is instantiated by the Action Executor, the specific weapon is bound to the variable myWeapon in the game code. The functions CheckPreconds() and CheckEffects() ensure that the action executes in a legal world state and that the effects asserted by the action are consistent with their representation in the plan operator, respectively. The action goes into the *Executing* state when the conditions are satisfied and the core execution code for the action is defined here in this state. In case of actions executing over durations of time, the action remains in this state throughout the duration of execution and the effects of the action are not asserted until the entire duration elapses and the action completes successfully.

A Camera Engine for Unreal

The Action Executor, a module within the Virtual Cinematographer written in UnrealScript, is responsible for managing execution of story and camera plans on the game engine. The high level camera directives that are generated by the camera planner are translated to geometric constraints for input to a constraint solver for camera placement. The translator accepts the abstract annotated action sequence and translates its contents into a set of geometric constraints characterizing the best shot-composition given a) the geometry of the world and b) the position of the shot's characters. The translator uses the cinematic rules to generate constraints on the composition of the camera's frame. These constraints are input to a constraint solver that computes a heuristically preferred position of the camera given the geometry of the world and the constraints that are required for a cinematic presentation of the shot. The constraint solver that is currently used in this work checks for occlusion from the desired camera location, specified by the action class. On failure to find an occlusion-free view, it shifts the camera position closer to the view target until there is no occlusion. For simplicity, this constraint solver does not search for all possible camera locations in the neigboring region of the desired shot.

Cinematic Rules

The rules of composition used in Darshak are described in the form of geometric constraints. The contents of the constraints are motivated by the attributes of the setting as specified by the designer and these are related to the actual properties of a camera. Each cinematic rule can be described through a collection of geometric constraints. For instance, a commonly used shot type is the close-shot, where the camera is positioned in such a way that the character's face is tightly fit into the shot frame. For this shot, the action class Close-Shot sets the ObjectOfAttention variable to the character it is filming, the shot-type to Shot_Close, direction to front and shot-angle to neutral. The ObjectOfAttention variable constraints the camera to focus on a specific object and is required to be set by each camera action class. The shot-type constraint is used to set the distance of the camera from the focus and can take various values that match the cinematic composition frames described by Arijon [2]. The Direction variable is used to constrain the camera's horizontal orientation with respect to the front of the object in focus. The Shot-angle specifies the vertical angle at which the camera looks at the object in focus. Each action class updates the constraints on the camera when it executes on the engine. A master constraint solver, described in detail in the next section, continuously sets the camera parameters to meet the current set of constraints. This master constraint solver is queried by the player's camera before updating the view on every tick cycle of the engine. Figure 3.15 shows the flow of control of the constraints on the game engine.

Constraint Solver

Given a director's shot request expressed in the form of visual composition constraints, the system must position the camera to satisfy the given constraints in the context of the given 3D virtual scene. The desired visual message or camera shot is expressed in the form of constraints on how subjects appear in the frame. The constraint solver then attempts to find values for each camera parameter so that all given constraints are satisfied. This process uses a heuristic that attempts to find values for camera parameters that are as near as possible to the parameters' optimal or preferred values. The camera constraint system developed in this work currently supports fifteen different types of constraints on either the camera's attributes or on how objects in the scene appear in the camera shot. Constraints include framing a subject in the camera's field of view, view a subject from a desired vantage angle, excluding or obscuring an undesired subject from the frame, avoiding occlusions, and a variety of constraints on how the subject's image is projected into the frame. Any constraint that may be applied to an object may also be applied to a designated part of the object's geometry. The scene and object geometry is represented using a combination of oriented bounding boxes and binary space partition trees. These geometric data structures are used to estimate projections of subjects in the frame, fraction of occlusion between subjects, and to keep the camera inside the walls and architectural features of a building interior.

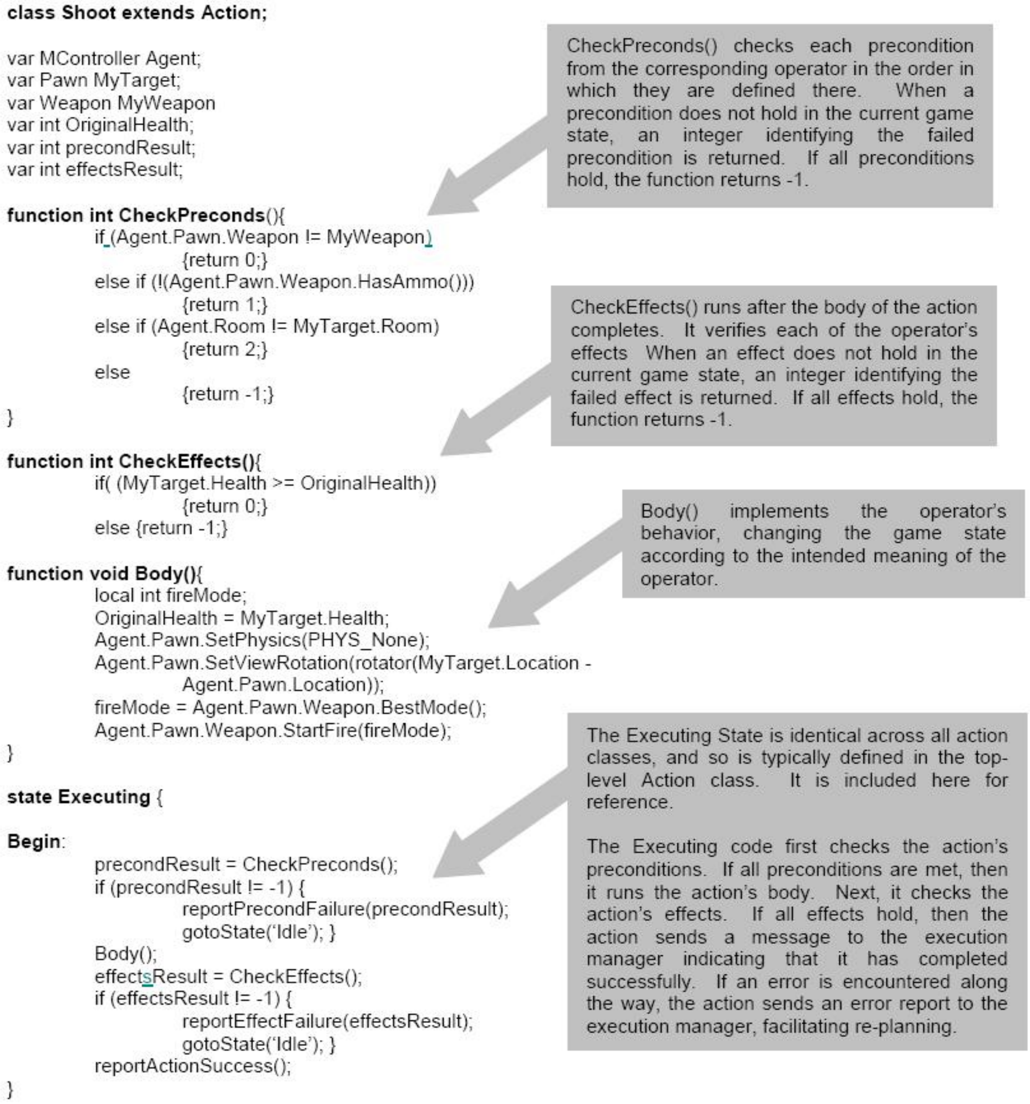
An execution monitor, which is implemented on the Unreal Tournament (UT) game engine, manages the execution of actions communicated by the server. It receives both the camera actions and story world actions. Camera control is implemented for a modification of the player type in the game ZFilmPlayer in the ZFilmGame game type in UT. The primitive camera action classes (close-up, medium-shot, long-shot, track-actor, pan-actor-to-actor, internal-shot) use the cinematographer object to set up constraints (Location, Rotation, Tracking, Lens) for the players camera. The players view is updated after getting recommendation for the camera position from the cinematographer, that takes into consideration the currently set up constraint values for updating the camera location.

3.6 Summary

It is possible, to a certain extent, to reason about discourse and story separately. I draw upon this distinction based on narratological notion of the fabula and the communicative media of a narrative (although narratologists's distinction between story and discourse is analytical and not directly concerned with a generative model of narrative). The Darshak system is based on this theoretical separation of the fabula and the discourse and defines the discourse layer as a distinct reasoning process separated from the fabula. Darshak's discourse planner operates on an input fabula, in the form of a plan data structure, and generates a sequence of camera directives to film the story. In order to reason effectively about the fabula plan, the discourse planner contains a rich representation of narrative patterns, the relationships between narrative patterns and popular visual idioms corresponding to narrative patterns, all represented as hierarchical plan operators along several levels of abstraction. Coherence, in a limited form, is established by exploiting a viewer's familiarity with popular cinematic idioms to communicate the story. Discourse generated as a result of Darshak's reasoning process is a combination of story actions and camera actions that are related through explicit temporal links. Having explicit temporal links ensures that the resulting visual discourse maintains temporal consistency during execution. This is important, as exact temporal scheduling of camera movements significantly affects viewer's comprehension and experience of the discourse.

The planning algorithm, described in this work, is a part of the Zocalo service-oriented architecture that provides a structured interface between Darshak's discourse planning algorithm and its execution environment. Zocalo provides tools that canfacilitate communication between the two modules by providing schema-based service management protocols and by providing tools that translate between different representations.

I have described an implementation of a constraint-based camera engine that includes a parameterized library of camera control actions. This design of the camera engine has several advantages: First, it is easily extensible. New compositions and shot constraints can be added easily through a specification of constraints that describe the composition. Second, it enables a straightforward translation from declarative abstract camera operators to code that manages the placement and movement of the camera. Lastly, it cleanly defines the abstract planning of camera actions and separates it from the details of manipulating shot properties and numerical geometric constraints.



```
class Shoot extends Action;

var MController Agent;
var Pawn MyTarget;
var Weapon MyWeapon
var int OriginalHealth;
var int precondResult;
var int effectsResult;

function int CheckPreconds(){
        if (Agent.Pawn.Weapon != MyWeapon)
                {return 0;}
        else if (!(Agent.Pawn.Weapon.HasAmmo()))
                {return 1;}
        else if (Agent.Room != MyTarget.Room)
                {return 2;}
        else
                {return -1;}
}

function int CheckEffects(){
        if( (MyTarget.Health >= OriginalHealth))
                {return 0;}
        else {return -1;}

function void Body(){
        local int fireMode;
        OriginalHealth = MyTarget.Health;
        Agent.Pawn.SetPhysics(PHYS_None);
        Agent.Pawn.SetViewRotation(rotator(MyTarget.Location -
                Agent.Pawn.Location));
        fireMode = Agent.Pawn.Weapon.BestMode();
        Agent.Pawn.Weapon.StartFire(fireMode);
}

state Executing {

Begin:
        precondResult = CheckPreconds();
        if (precondResult != -1) {
                reportPrecondFailure(precondResult);
                gotoState('Idle'); }
        Body();
        effectsResult = CheckEffects();
        if (effectsResult != -1) {
                reportEffectFailure(effectsResult);
                gotoState('Idle'); }
        reportActionSuccess();
}
```

CheckPreconds() checks each precondition from the corresponding operator in the order in which they are defined there. When a precondition does not hold in the current game state, an integer identifying the failed precondition is returned. If all preconditions hold, the function returns -1.

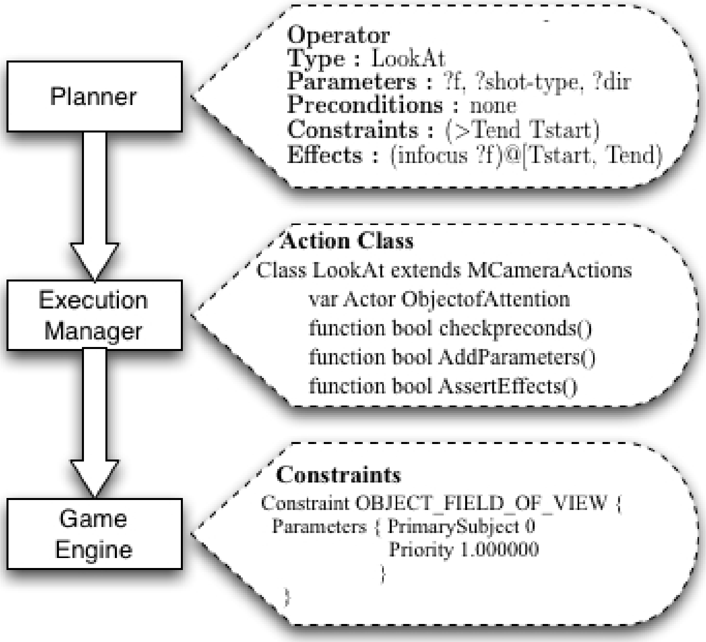
CheckEffects() runs after the body of the action completes. It verifies each of the operator's effects When an effect does not hold in the current game state, an integer identifying the failed effect is returned. If all effects hold, the function returns -1.

Body() implements the operator's behavior, changing the game state according to the intended meaning of the operator.

The Executing State is identical across all action classes, and so is typically defined in the top-level Action class. It is included here for reference.

The Executing code first checks the action's preconditions. If all preconditions are met, then it runs the action's body. Next, it checks the action's effects. If all effects hold, then the action sends a message to the execution manager indicating that it has completed successfully. If an error is encountered along the way, the action sends an error report to the execution manager, facilitating re-planning.

Figure 3.13: Example of an action class definition in UnrealScript. To create new action classes, developers can create sub-classes of the abstract action class and implement the essential functions for checking conditions in the world and physically asserting effects of the action class on the game state.



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

```
Action Class
Class LookAt extends MCameraActions
        var Actor ObjectofAttention
        function bool checkpreconds()
        function bool AddParameters()
        function bool AssertEffects()
```

```
Constraints
Constraint OBJECT_FIELD_OF_VIEW {
  Parameters { PrimarySubject 0
                    Priority 1.000000
                  }
}
```

Figure 3.14: Representation of Camera Actions in Different Modules.

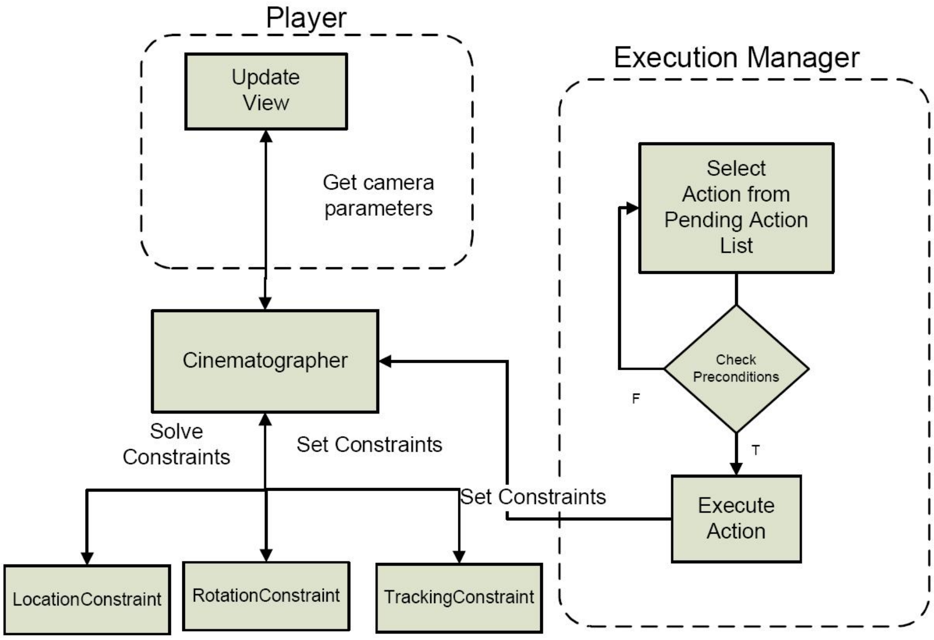


Figure 3.15: Flow of control for the execution manager on the game engine.

Chapter 4

Experimental Evaluation

The Darshak system generates narrative discourse in the visual medium based on an input fabula plan produced by a story generation system. While one of the contributions of this work is the representation of visual discourse in a planning formailsm, another contribution is the overall functionality of Darshak. The Darshak system is capable of generating story visualizations that are coherent, and follow acceptable cinematic conventions. The goal of the system is to ensure that the the cinematic discourse that it generates is coherent to viewers. Coherence, however, is not an attribute that can be readily measured, since it is a subjective attribute of the story as percieved by viewers. Thus, in order for us to determine whether or not visualizations generated by Darshak are coherent, is is necessary to take measure of the perceived attributes of the story by viewers. Empirical evaluation of such a subjective metric is challenging due to the following reasons:

- Viewers rarely share a common definition of coherence. They cannot be asked directly to give judgement on coherence.
- Different Viewers report different levels or measures of perceived coherence, making it difficult to map the values that they report directly to a unifrom scale.
- While discourse certainly affects coherence, coherence is also a property of the fabula plan itself. The audience perception of the story, rendered in a communicative medium, is altered by the transformation of the fabula plan into the communicative elements. Any evaluation of the communicative elements must take into account this inherent coherence in the fabula itself.

- Viewers' subjective impressions of the story, for instance, their likes and dislikes about the plot, are also affected by the quality of character dialog and animations. It is difficult for subjective surveys to control for these effects.

To address these challenges, one way of empirically evaluating discourse generated by Darshak is by indirectly evaluating the cognitive representations formed by the audience members during the presentation of stories it creates. While the cognitive representations of the story are not directly accessible to the audience, their structure can be inferred by measuring audience members' performance on various cognitive tasks related to the story such as recall [13] or question-answering [14]. Measurements of cognitive tasks have been found to be successful in measuring perceived coherence in fabula plans generated by planning algorithms [60]. In the following section, I first describe the basis for the use of cognitive models for empirically evaluating the Darshak system, then present the design and experimental methodology for evaluating this system. Finally, I present the analysis of the experiments and discuss the data produced through them.

4.1 Background

To empirically evaluate a viewer's perception of the coherence of the story, I based my evaluation on the QUEST model of question-answering developed by Graesser, Lang and Roberts [14]. In the QUEST model stories are represented as conceptual graph structures which contain concept nodes and connective arcs. These graphs are called QUEST Knowledge Structures (QKSs). They describe the reader's conception of narrative events and their relationships. Nodes and arcs in a QKS structure are based on their purpose in the narrative. For instance, if nodes A and B are two events in a story such that A causes or enables B then the events A and B are represented by nodes in the QKS graph and they will be connected by a Consequence type of arc.

Techniques used by Graesser et. al. to validate the QUEST model are based on goodness-of-answer (GOA) ratings for question-answer pairs about the story given to readers. Goodness-of-answer ratings are judgements that readers provide about the quality of answers in question-answer pairs after reading the stories on which they are based. GOA ratings are discrete values ranging from bad to good. The questions and answers in the pair are selected from goal or event nodes from the QKS structure. This structure is used by QUEST predictor variables to predict GOA ratings. To validate QUEST's model of question-answering, GOA ratings obtained from the viewers are compared to ratings predicted by the QUEST model. In the QUEST experiments, the purpose of this model was to show that viewers build cognitive representations of the story they see that capture certain relationships between events in a story and, more importantly for Graesser and his colleagues, to show that the QUEST procedures traversed these QKSs in a manner that produced similar answers to those that would be produced by humans.

Unlike Graesser and others, I make use of these techniques (employing QKSs and GOA ratings) not to measure inferences made by viewers, but rather to validate the mental models formed by viewers upon watching the cinematics produced by Darshak. As described in more detail below, I use an automated planning system first to generate a story plan that is then provided to Darshak as input. Darshak then creates a cinematic visualization of the story which human subjects then view. After they watch the resulting video, the subjects are prompted to provide goodness-of-answer ratings to a set of automatically generated question/answer pairs relating to the story plan they viewed. Procedures defined by the QUEST system are run to determine expected goodness-of-answer ratings for the question-answer pairs shown to subjects and the subjects' responses are compared to these automatically generated ratings to determine how well the subjects understood the story based on the particular visualization they watched.

QUEST supports questions of types why, how, when, enablement, and consequence. For example, consider the story in Table 4.1, taken from Graesser, Lang and Roberts ([14], Fig 1), and the corresponding QUEST knowledge structure (QKS) shown in Figure 4.2.

The links in Figure 4.2 represent the different types of relationships between events and character goals within a story.

- Consequence(C): The terminal event node is a consequence of the initiating event node.
- Reason(R): The initiating goal node is the reason for the terminating event node.
- Initiate(I): The initiating event node indicates a terminal goal node.
- Outcome(O): The terminal event node is the outcome of the initiating goal node.

> Once there was a Czar who had three lovely daughters. One day the three daughters went walking in the woods. They were enjoying themselves so much that they forgot the time and stayed too long. A dragon kidnapped the three daughters. As they were being dragged off, they cried for help. Three heroes heard the cries and set off to rescue the daughters. The heroes came and fought the dragon and rescued the maidens. Then the heroes returned the daughters to their palace. When the Czar heard of the rescue, he rewarded the heroes.

Figure 4.1: Example story used to evaluate the QUEST model of question-answering

- Implies(Im): The initiating event node implies the terminal event node.

Following question based on node 5 in Figure 4.2 illustrates the QUEST model of question-answering pertaining to the story in Table 4.1: ”Why did the daughters stay in the woods too long?”. The question is a query about node 5 and some of the possible answers are shown below:

A. Because the daughters forgot the time (node 4).

B. Because the dragon kidnapped the daughters (node 7).

C. Because the daughters were walking in the woods (node 18).

One of the predictors defined in the QUEST model to determine legal answer nodes is the arc search procedure. This procedure starts at the queried node and traverses the QKS graph till it finds a path to the answer node. If no such path exists, then the answer node is deemed illegal. In this example, nodes (A) and (C) are legal answers. Of these answers, (A) is preferred as node 4 has shorter structural distance from the question node 5. According to the arc search predictor, the predicted GOA rating of node 4 is higher than that of node 5. Computations like these provide specific numerical values for goodness-of-answer for different question-answer pairs used in my experiment. These values were compared to average GOA ratings obtained from viewers’ responses to requests for the same questions and answers.

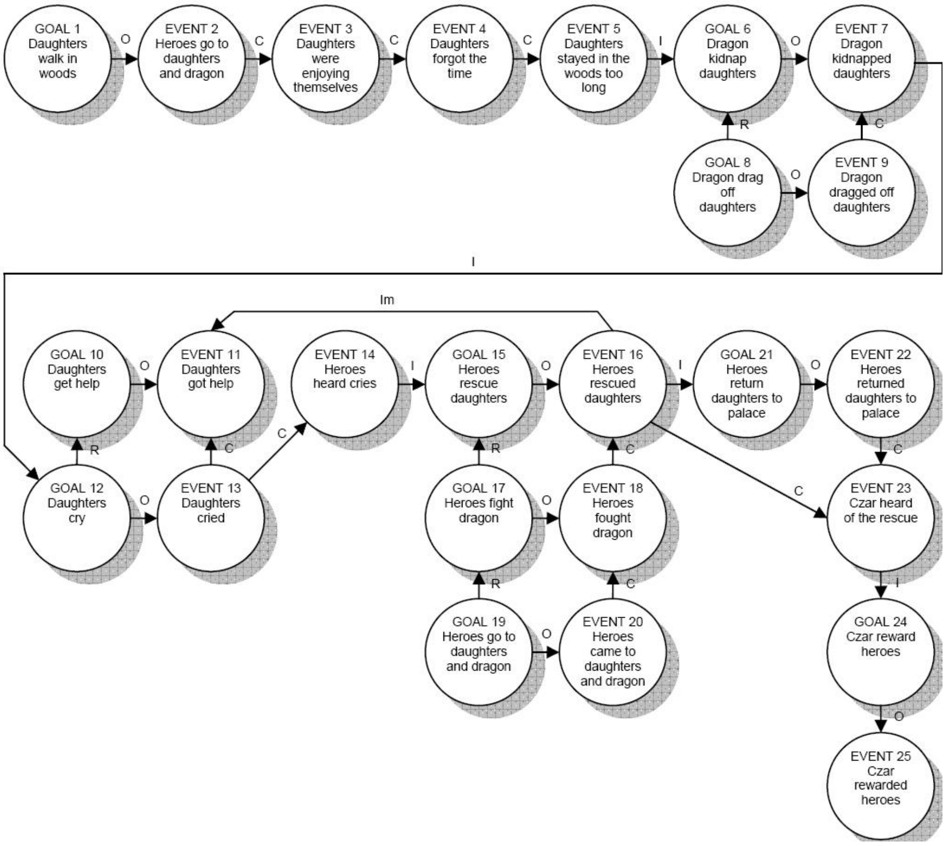


Figure 4.2: QKS structure corresponding to the QUEST example story

Christian and Young [60] have shown that stories generated in the form of DPOCL plan structures can serve as an effective proxy for a user's understanding of the story. In their experiment, plan data structures representing the story were first automatically converted to QKS structures - graph structures representing story actions into nodes and relationships between them as edges - defined by Graesser et. al., described in the previous section. The algorithm for converting a DPOCL plan data structure to the corresponding QKS structure is shown in Table 4.3.

In their work, predictor variables were calculated from the resulting QKS for predicting the goodness of answer (GOA) rating - the measure of goodness of answer for a question/answer pair related to the events in the story. These GOA ratings were compared against data collected from 15 subjects who watched a video of the story filmed using static master-shots. That is, in the work by Christian and Young, camera shots were selected explicitly to provide as little affect on measures of coherence as possible. The experiments measured the effectiveness of the QKS generated from a DPOCL plan structure in predicting the Goodness of Answer (GOA) ratings.

GOA ratings in the models proposed by Graesser et.al. are determined through QUEST's predictor variables. Specifically, the three predictors that are correlated to the GOA ratings are *arc search*, *constraint satisfaction*, and *structural distance.* Users who participated in the experiments performed by Christian and Young were shown a video of a story through fixed viewpoints in a virtual world. They were then given question/answer pairs from the story and were asked to rate the quality of answers. These results were compared to the GOA ratings predicted by the QKS structure based on the DPOCL plan used to drive the characters in the story world. 15 participants were randomly assigned to three groups with different categories of questions of the forms: how, why and what enabled.

Let n be the number of top-level goals of plan P and let m be the number of steps in P.

1. Create a total ordering o for the m steps in P that is consistent with the ordering constraints of P.
2. For each goal g_i of plan P for i=1, ..., n, convert g_i into a goal node G_i.
3. For each plan step s_j for j=1, ..., m starting wiht the last step in o.
    a. Convert s_j into a goal node $_j$ and an event node E_j
    b. Link G_j to E_j with an outcome arc.
4. For each causal link in P connecting two steps s_1 and s_2 with condition c, connect the event node E_1 to the event node E_2 with a consequence arc.
5. For each causal link $<s_1,p,q,s_2>$ in P connecting two steps s_1 and s_2, connect G_1 to G_2 with a reason arc.

Figure 4.3: Algorithm for converting a POCL plan data structure to corresponding QKS.

In contrast to the work described in this dissertation, Christian and Young measured the effective comprehension of plan-based narrative representations *at the story level.* Their experimental design is similar to ours, however, in that they employ a cinematic model to convey the underlying story structure to their subjects. While their selection of fixed, master-shot style shots to convey the story helped to minimize discourse effects on comprehension, it is clear that, in general, selection of viewpoint in cinematics is a part of a director's intentional communicative action designed and understood as an attempt to influence the comprehension of viewers. While Christian and Young's work is the closest to mine with respect to the empirical evaluation of automatically generated narratives, direct comparison between their work and Graesser's work cannot be made because the text used in Graesser's experiments was hand-crafted with a goal of influencing readers. Discourse strategies, like order and structure of sentences and choice of words, that were present in text used in Graesser's experiments were not utilized by Christian and Young, as their approach used fixed camera angles. In this dissertation, I argue that intentionally constructing visual discourse for communication improves the communication effectiveness of computational models. I seek to specifically evaluate this claim through the experiment described below.

4.2 Purpose

In this experiment, my method adapts techniques used in experiments carried out by Christian and Young [60]. Christian and Young's experiments were carried out to evaluate the effectiveness of computational models of narratives. They were themselves based on the design used in Graesser et.al.'s [14] experiments. I designed my experiments to test whether the quality of visualizations that are mapped to the QKS structure affect comprehension. Specifically, I sought to determine whether Darshak's intentionally generated visualizations based on communicative goals improved comprehension compared to fixed viewpoints that were used in creating videos by Christian and Young. If story visualizations that were intentionally constructed using Darshak's representation of cinematic conventions and its plan construction algorithm satisfy a set of cinematic communicative goals, then the GOA ratings provided by human subjects viewing the resulting cinematics would be higher (that is, users would gain a better understanding of the story) compared to GOA ratings provided by subjects watching visualizations not informed by the context of the stories.

4.3 Method

4.3.1 Participants and Design

The experiments were carried out on 30 participants, primarily undergraduate and graduate students from Computer Science. I used 2 stories (S1 and S2) and three visualization strategies for each story (V1-fixed camera, V2-randomly selected camera angle, and V3-Darshak driven camera) yielding 6 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 6 groups (G1 to G6). Each participant was asked to rate question-answer pairs of three forms of how, why and what enabled for videos representing two unique story and visualization pairs.

To address one of the challenges mentioned earlier, of accounting for the inherent coherence in the fabula, and to account for the effects of watching videos in order, I chose to follow the Youden squares design to distribute the groups among the 6 treatments. Assuming a continuous response variable, the experimental design, known as a Youden square, combines Latin Squares with balanced, incomplete block designs or 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 BIBDs. Since both group and visualization appears 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 effct, thus simplifying computation.

Table 4.1: 2x3 Youden squares design for the experiment. G1 through G6 represent 6 groups of participants with 5 members in each group. They are arranged so that each story and visualization pair has a common group for other visualizations

Visualization	**Master Shot**	**Over The Shoulder**	**Darshak**
Story 1	G1, G4	G2, G5	G3, G6
Story 2	G5, G3	G6, G1	G4, G2

4.3.2 Materials

Six videos, one for each of the three variants for the two stories, were shown to the participants. The stories for this experiment consisted of 15 steps corresponding to 70 QKS state/event-goal nodes. These numbers were chosen in order to keep the story lengths comparable to the earlier experiment. The algorithm for converting plan data structures to QKS graphs is identical to that used in Christian and Young and is shown for reference in Figure 4.3. One story used in the experiment is shown in Figure 4.4 and the corresponding QKS graph is shown in Figure 4.5.

Of the 70 QKS nodes, 10 questions from story 1 and 13 questions from story 2 were generated from randomly selected and converted to one of the three question types supported by QUEST: how, why, and what enabled. For each of the 10 questions, approximately 15 answer nodes were be selected from nodes that were at a structural distance <3 in the QKS graph generated from the story data structure. These numbers were chosen to have similar magnitude to the previous experiments, for better comparison.

4.3.3 Apparatus

The videos were shown to participants on a personal computer with a 21 inch viewing screen. The survey was conducted on an online form in which the users selected their GOA response by checking a radio button for each question-answer pair. All question-answer pairs were shown to the viewers on a single browswer page that could be scrolled. Post-experiment interviews were conducted privately by the observer and answers recorded on paper.

4.3.4 Procedure

The entire experiment was carried out in a single session for each participant. Total time for a single participant was between 30 and 45 minutes. Each participant went go through three stages during the experiment. The three stages are briefly described in the following paragraphs.

Instructions - 15 minutes. 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. The instructions, consent form, and other

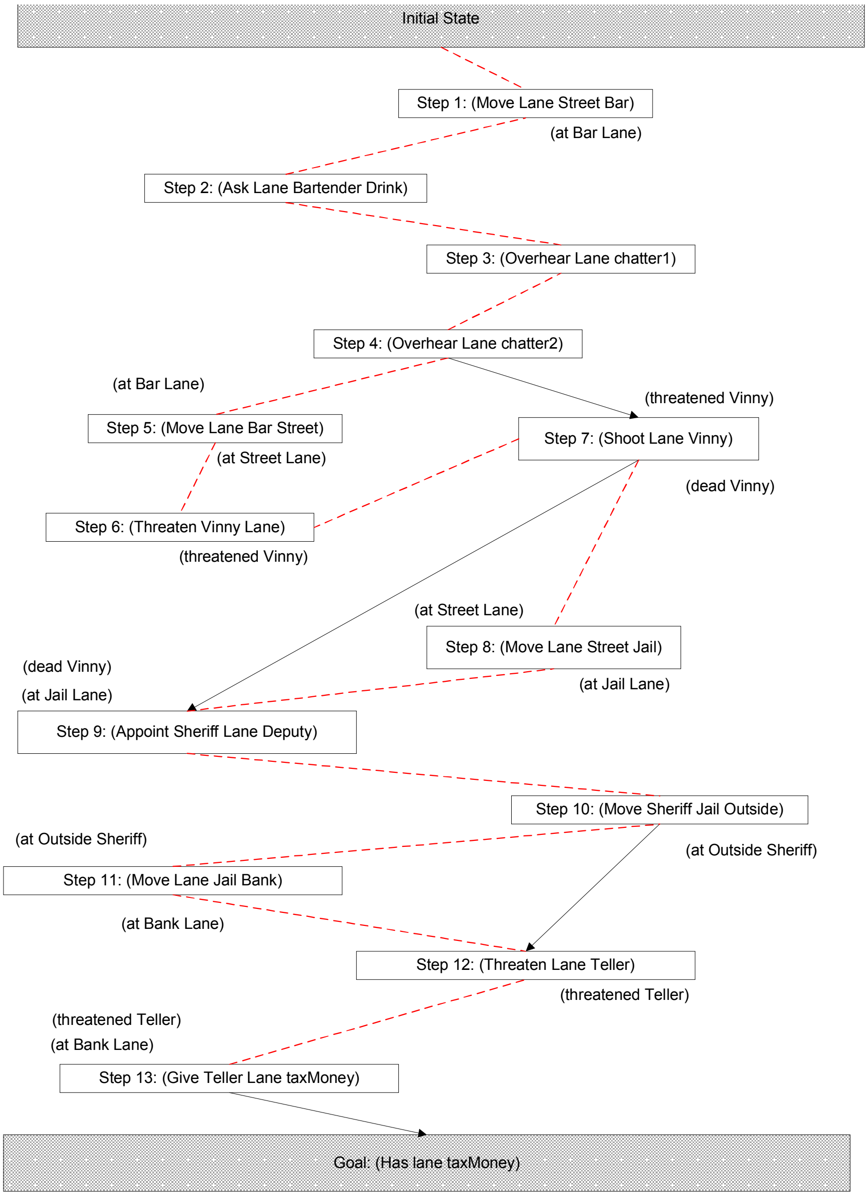


Figure 4.4: Simplified pictorial representation of the example story plan input to the planner. Boxes represent story actions, red dotted lines indicate ordering links, and solid black arrows denote causal links

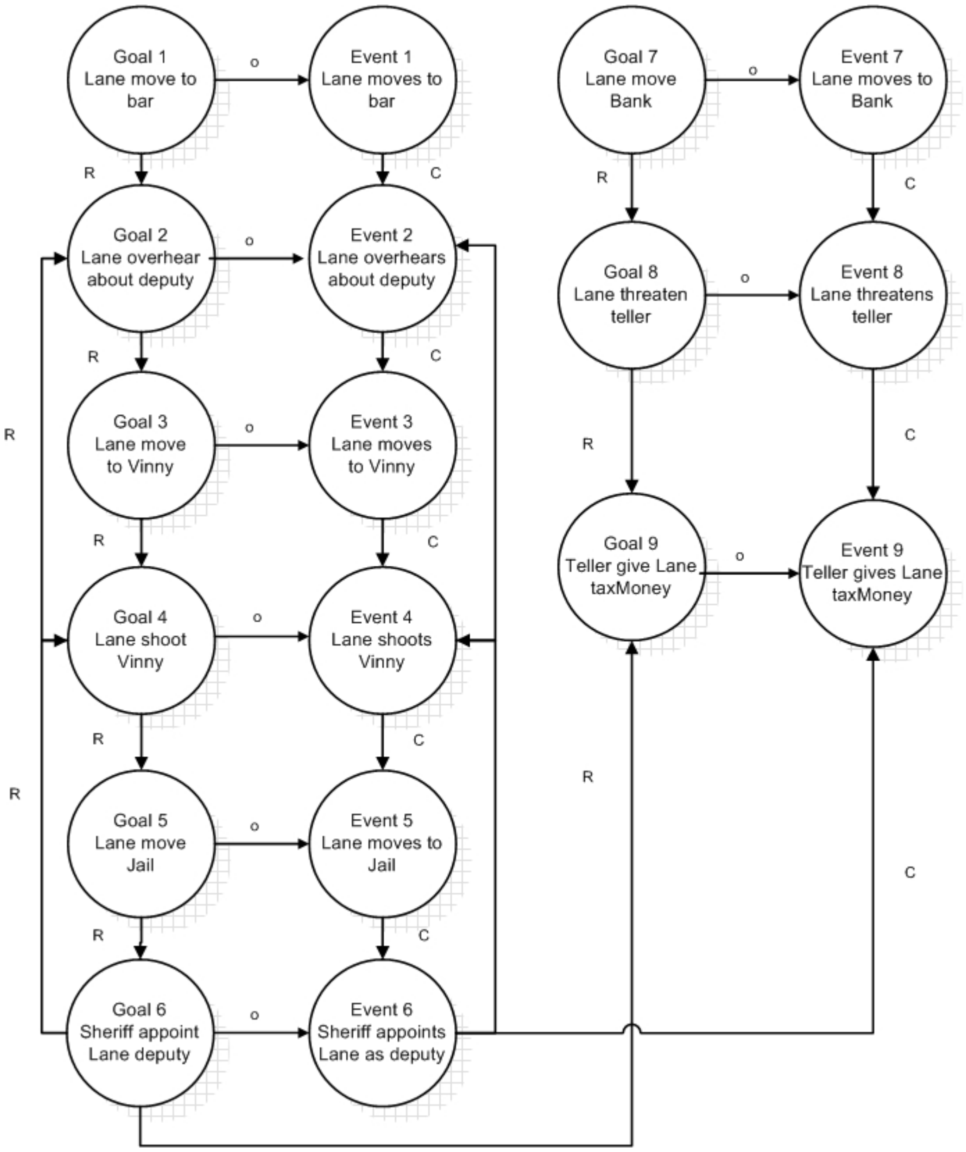


Figure 4.5: QKS structure for one of the stories from the experiment. Goal and Event nodes from the story are represented by circles. Relationships between nodes is indicated by arrows which are labeled respectively as Reason(R), Consequence(C), and Outcome(O).

materials are given for reference in Appendix 5.1.

Viewing - 15 minutes. After their briefing, users watched a video of one story with a particular visualization according to the group assignment (Table 4.1).

Filling out Questionnaire - 30 minutes. Subjects were then asked to provide GOA ratings for 10-12 question-answer pairs relating to the video they had just watched. 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. Next, they watched a second video with a different story and visualization followed by a questionnaire about the second story. This questionaire asked subjects to provide the same GOA ratings for a set of questions relating to the second video. 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.

4.4 Results

Statistical analysis was carried out for the effects of three predictor variables viz. arc search, structural distance, and constraint satisfaction. As a point of reference, Graesser's analysis in his earlier, text-based experiments shows a significant positive coefficient for the first two and a significant negative coefficient for the last variable. In my experiement, I observed relatively higher values for these coefficients on the videos generated by Darshak compared to Master-Shot(MS) and Over-the-Shoulder(OTS) approaches.I further observed the values of the coefficients to be significantly closer to Graesser's original text based experiments than the MS and OTS approaches.

Previous experiments on the use of plan-based models and narrative comprehension established plan structures as effective representations of a user's mental model build during the process of story understanding. The experiments proposed here provide further insight into the effectiveness of computational models of narrative representation by explicitly analyzing the effects of different discourse structures used to convey the same stories.

The mean overall GOA ratings recorded for the two stories are shown in Table 4.2 along with the standard deviations. These distributions of GOA scores do not present any problem for statistical 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 results of T-test analysis indicate that GOA ratings for V1 and V3 are significantly different for both stories (p=0.03 and p=0.05). While the ratings are not significantly different for comparisons between V2 and V3 (p=0.09 and p=0.7) and between V1 and V2 (p=0.1 and p=0.7). This could be attributed to the inherent cohernce of the fabula and similar visualizations for V2 and V3. The GOA charts shown in Figures 4.6 and 4.4 indicate on preliminary observation that the GOA ratings for V1(Master Shot) and V3(Darhsak) are significantly closer than V2(Randomly selected 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 visualization. An

Table 4.2: Mean GOA ratings (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)

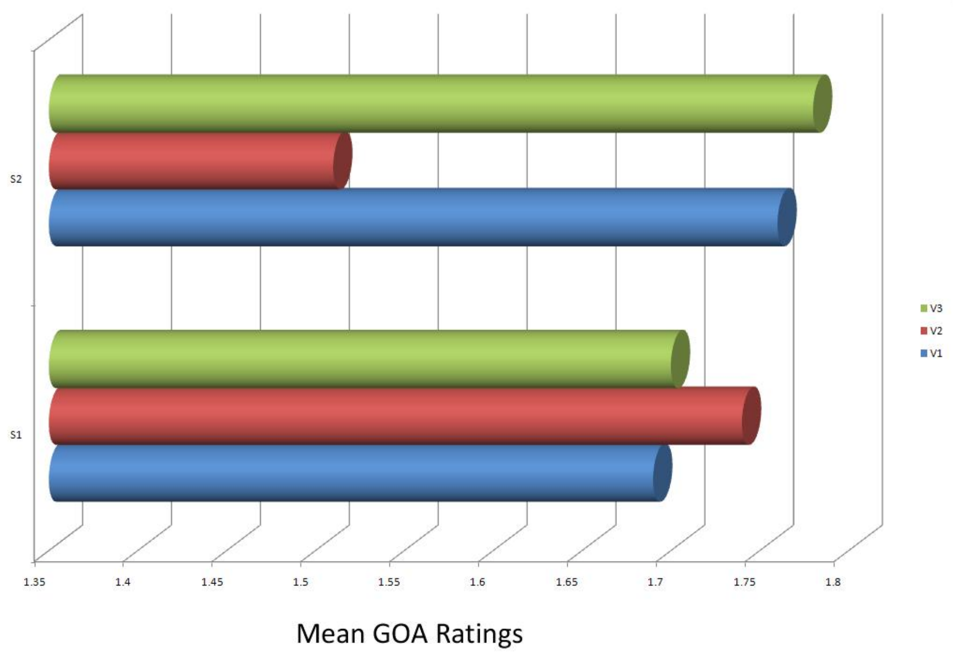
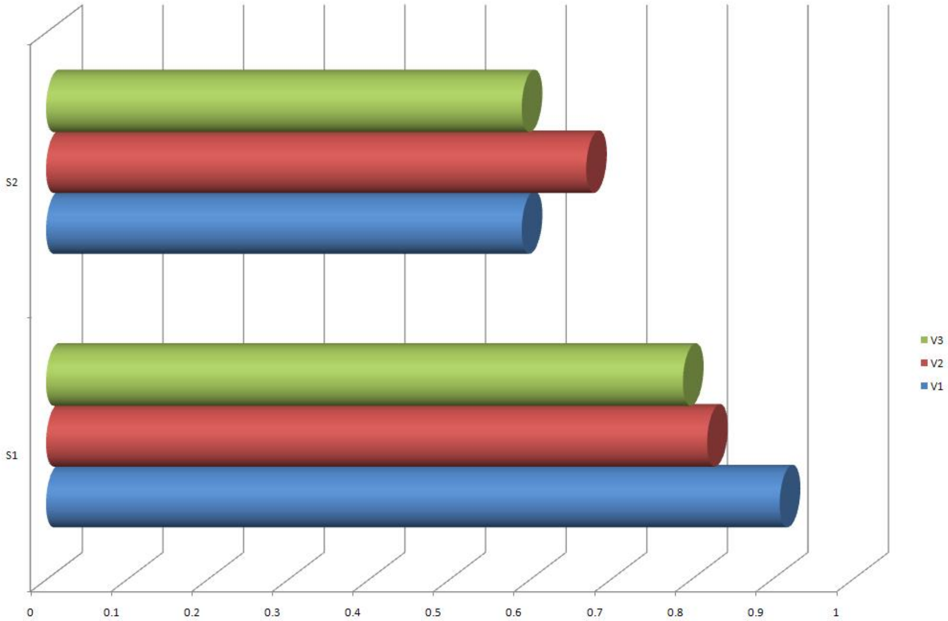


Figure 4.6: Mean GOA rating



Standard Deviations in GOA ratings

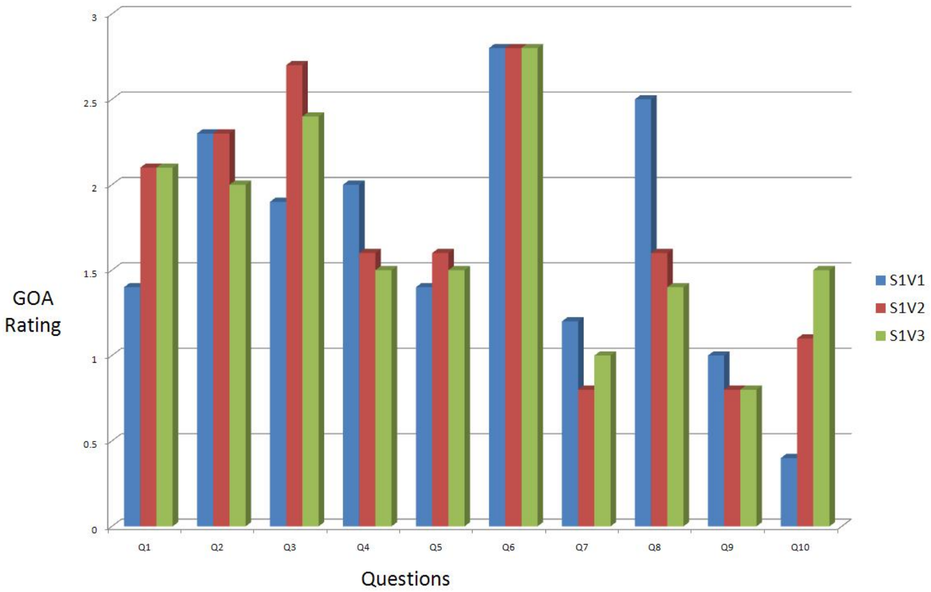
interesting observation for V2 is that in story 2 the mean GOA ratings are significantly lower than the other two treatments with a significantly high standard deviation. These figures support the intuition that participants form their own interpretation of events in the story while looking at shots that are randomly selected leading to the wide disparity in ratings in going from story 1 to story 2. While mean ratings provide an overall idea of the participant's responses, it is interesting to observe disparity in GOA ratings for individual questions across different visualizations. Figure 4.4 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 significant difference 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:

Q1 Why did Lane challenge Vinny? A. Because he wanted to kill Vinny.

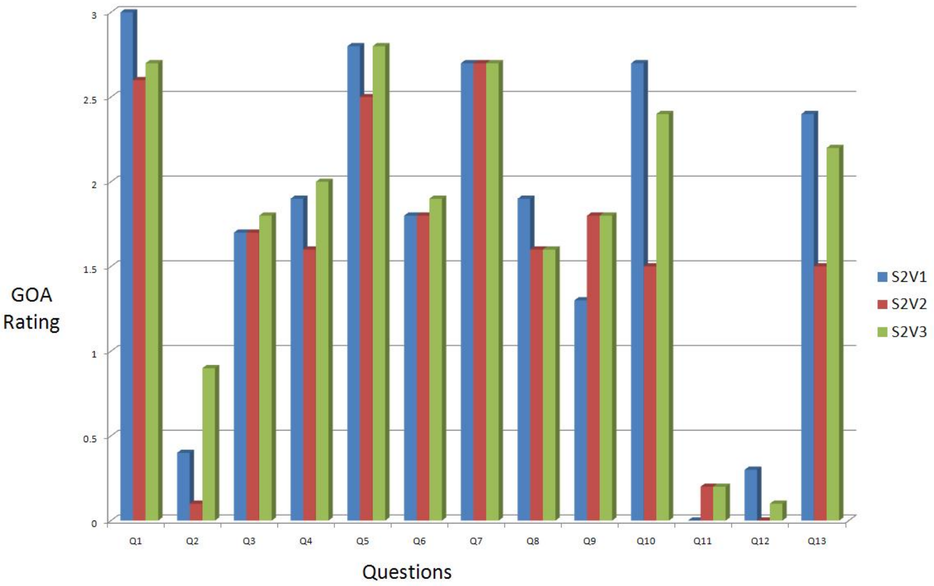
Q8 Why did Lane challenge Vinny? A. Because Lane wanted to steal tax money.

Q10 Why did Lane meet Sheriff Bob? A. 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. This is an interesting case where users who were familiar with the gunfight setting chose to label the challenge as being an important step in killing Vinny. In a master-shot the gunfight sequence was not even recognized as a gunfight by most participants. This was one of the story events that was explicitly mentioned in the qualitative feedback. Further analysis on the 'why' type of questions on other nodes is needed to specifically determine the effects of visualization strategies on GOA ratings in detail. Figure 4.4 shows the average ratings for each question for the second story. The interesting responses are the ones that have a significant difference in mean ratings across different visualizations. In this story, unlike 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).



4.5 Qualitative Survey

Post-experiment interviews were carried out after participants completed the rating forms for both stories. During these interviews the participants were asked the following questions:

1. Which story did you like ?
2. Which story was better in terms of presentation ?
3. Did you find any unexpected information about the story from the survey question/answers ?
4. Did you change a previously recorded answer after looking at another question/answer pair ?
5. Which story had stronger characters ?

The main purpose of these questions was to get additional information about metrics that have not been considered in previous approaches but may play a role in analysis of GOA ratings. The data collected from this survey provides insight into a possible extension of the cognitive model of story understanding that takes into account features of discourse that are not currently represented.

4.6 Summary

This chapter describes the evaluation study carried out on visual discourse generated by Darshak. The study determines whether viewers perceive a story visualized through camera shots generated by Darshak to be more comprehensible and coherent. For quantitatively measuring comprehension the experiment was modeled after established cognitive story understanding experimental techniques and models. The study consisted of 6 treatments, one each for a story and a visualization pair, for two stories and three visualizations in a Youden squares design. Participants were divided into 6 groups of 5 participants each. Each treatment was allocated two groups with one group getting the treatment as the first treatment and the other group getting that treatment as the second one. One of the three visualizations shown to the viewers was generated by the system. Users were asked to rate question-answer pairs on a 4 point scale (Bad, Somewhat Bad, Somewhat Good, Good). The mean ratings of the viewers were compared to predicted GOA ratings from the QKS structures. It was found that GOA ratings were indeed affected by visualization strategies. An analysis of standard deviations in the ratings showed that system generated visualizations received uniform ratings from the viewers while other visualizations showed significant variability. I also found significant improvement in GOA ratings for questions related to actions that were shot using cinematic patterns. Overall the results are encouraging and support the claim that different visualization strategies do affect comprehension measured by GOA ratings. Furthermore, these results support the claim that visualizations generated by Darshak are coherent.

Subjects, who participated in the experiments, were also asked to provide subjective feedback about the stories and visualizations. From the subjective responses, a majority of the subjects preferred system generated videos to the other two videos. Most users reported that visualization affected 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 statistically measure, the uniformity in responses across groups points at the importance of having intentional camera control compared to static pre-defined camera shots and randomly generated shots.

Chapter 5

Conclusions and Future Work

In this chapter, I conclude with a summary of the major contributions and findings of the work presented in this dissertation. This dissertation presents Darshak, a system for representing and reasoning about the communication of narratives through cinematic camera control within a 3D graphical environment. Darshak uses a planning approach to constructing visual scenes of narratives. Scenes generated by Darshak are coherent, follow acceptable cinematic conventions, and maintain focus of the viewer on salient elements of the narrative. Current intelligent camera planning systems do not attempt to inform the selection of camera shots from the underlying narrative context. In this work, I have provided a representation of cinematic idioms that enables a planning algorithm to reason about selection of camera parameters based on the narrative context. Narrative generation systems have focused on rich representations of the fabula and used text as the communication medium. Recent story generation systems have been implemented on interactive graphical environments. Such environments provide the opportunity to procedurally construct visual discourse through camera control for telling the unfolding narrative. Multi-sentential discourse generation systems have been successful in generation of coherent text for effectively achieving communicative goals. Such text-based systems have also been used in generation of discourse for telling automatically generated narratives. Recent interactive storytelling systems are based in graphical environments. While text-based systems have been evaluated through cognitive evaluations for coherence, the effect of visuals in understanding of stories has not been evaluated. Specifically, Darshak seeks to solve the following problem that addresses the challenges mentioned above.

> **Problem Statement:** Given a story, in the form of parameterized set of actions with causal and temporal relationships between actions, and a set of communicative goals, construct a schedule of communicative shots and create a visualization of the story such that it is coherent, comprehensible, and follows acceptable cinematic conventions.

I have established saliency, coherency, and temporal consistency as significant properties of a solution to the above problem. Darshak's representation of film idioms, extensions to discourse planning algorithm, and integration with a geometric constraint solver produces visual scenes that: depict salient elements of the story, are coherent, and follow acceptable cinematic conventions. These properties of the system lead to generation of cinematic discourse that has been empirically evaluated to be coherent and following acceptable cinematic conventions. Through a solution to this particular problem Darshak makes specific contributions to the following fields.

- **Geometric Camera Control:** Darshak contains an implementation of film shots that are represented by a collection of constraints. Darshak's representation of film idioms extends constraint based techniques by defining a relationship between low-level geometric constraints and abstract cinematic idioms.Prior approaches to modular constraint based camera control are extended with a richer representation of modules through their association with planning operators with an explicit formalization of their effects on the cognitive model of a viewer.
- **Intelligent Narrative Generation:** In intelligent narrative generation, less emphasis has been placed on the visual communicative effects of the story that is presented to the user. This work establishes that visual communicative actions can be effectively represented through communicative plan operators and utilized for deliberate planning for cinematic storytelling.
- **Discourse Generation:** Darshak extends existing approaches to discourse planning by introducing cinematic idioms and camera shots as communicative operators. I have extended discoure planning algorithms by providing explicit representation and reasoning of temporal relationships between discourse actions and story actions.
- **Evaluation of Narrative Generation Systems:** Through evaluation of Darshak I have extended the evaluation techniques used for evaluating cognitive and computa-

tional models of story to specificallly observe the effects of various discoures strategies used to tell the story in a visual medium. Prior work on evaluating cognitive models measured the effectiveness of the model in the textual medium but did not take into account various discourse strategies. Prior work on evaluating computational models of narrative was based on visuals but did not take into account various visual communicative strategies. 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.

5.1 Future Work

The work presented in this dissertation is a step towards developing intelligent systems that can generate effective cinematic discourse to tell stories. Such systems can take advantage of the graphical complexity and procedurality of graphical engines by constructing powerful movies based on cinematic knowledge. This dissertation presents an end-to-end system with representation of narrative patterns and cinematic idioms, a planning algorithm as a reasoning mechanism, and an execution environment for realizing cinematic discourse. The results of this research paves way for several rich research problems in representation of story and discourse, reasoning algorithms for generation of multi-modal discourse, and cognitive models of viewer comprehension in visual discourse.

The formalization of cinematic idioms and narrative patterns in this dissertation covers a small subset of the many documented and undocumented cinematic conventions. Appendix 5.1 lists some of the documented conventions. The idiom representation could be enriched by explicitly representing dramatic parameters like emotion, character personalities, and conflict. Undocumented conventions can be mined from the wealth of information available on annotated movie scripts [61].

Further investigation is needed to understand the cognitive effects of camera sequences, like suspense and surprise [62], on viewers. The current cognitive models are models of the fabula and measure the effects of the discourse in a limited form. In the qualitative surveys carried out during the evaluation experiments, several subjective questions (see Section 4.5) were asked to viewers after the quantitative GOA survey. The motivation for asking these specific questions were varied. The first two questions were asked to judge the participants preferences about the type of visualizations. The third question was to detect if the text contained in question-answer pairs did contain information that was not available from looking at the visualization itself. This was done to get an insight on whether information from the question-answer text influenced the inferences that viewers made about the story. The fourth question was asked to relate the ratings with the confidence of the participants. This is an interesting metric, not discussed directly in previous approaches, that needs to be investigated further.The fifth question was asked to see if visualization indeed affects the participant's perception of character personality and if it is an important factor in engagement as well as understanding. As stated earlier, the qualitative survey was meant to gather data on aspects of the experiments that were not directly available from the quantitative questions. This data would also inform experimental design and methodology for future experiments.

Finally, a number of applications [59, 63] have started using the system develped in this dissertation as intelligent tools to support cinematic creativity. Further development and refinement of such automated and mixed-initiative tools could support and enhance cinematic creativity. Such tools, with extensions for designing interaction in addition to story content, could prove valuable for authors of interactive narrative systems and support development of this new creative medium.

Bibliography

[1] Alfred Hitchcock. The rope. Movie, 1948.

[2] Arijon D. *Grammar of the film language.* Silman-James Press, 1976.

[3] Mascelli Joseph. *The Five C's of Cinematography.* Cine/Grafic Publications, 1970.

[4] William H. Bares, Joel P. Gregoire, and James C. Lester. Realtime constraint-based cinematography for complex interactive 3d worlds. In *AAAI/IAAI*, pages 1101–1106, 1998.

[5] Steven M. Drucker and David Zeltzer. Camdroid: A system for implementing intelligent camera control. In *Symposium on Interactive 3D Graphics*, pages 139–144, 1995.

[6] Alexander Hornung, Gerhard Lakemeyer, and Georg Trogemann. An autonomous real-time camera agent for interactive narratives and games. In *Intelligent Virtual Agents*, 2003.

[7] Halper Nicholas, Helbing Ralf, and Strothotte Thomas. A camera engine for computer games: Managing the trade-off between constraint satisfaction and frame coherence. In *proceedings Eurographics*, volume 20, pages 174–183, 2001.

[8] G. Prince. *A Dictionary of Narratology.* University of Nebraska Press., Lincoln, NE, 1987.

[9] Chatman S. *Story and discourse: Narrative Structure in Fiction and Film.* Cornell University Press, Ithaca, New York.

[10] M. Bal. *Narratology: Introduction to the Theory of Narrative.* University of Toronto Press., Toronto, 1997.

[11] Edward Brannigan. *Projecting a Camera.* Routledge, New York, 2006.

[12] Mark C. Langston, Tom Trabasso, , and Joseph P. Magliano. A connectionist model of narrative comprehension. In *Understanding Language Understanding.* MIT Press, 1999.

[13] Paul V. Broek. The effects of causal structure on the comprehension of narratives: Implications for education. *Reading Psychology*, 10(1):19–44, 1989.

[14] A. C. Graesser, K. L. Lang, and R. M. Roberts. Question answering in the context of stories. *Journal of Experimental Psychology:General*, 120(3), 1991.

[15] Mark O. Riedl and R. Michael Young. An intent-driven planner for multi-agent story generation. *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, New York*, 2004.

[16] Drucker Steven and Zelter David. Intelligent camera control for graphical environments. In *Graphics Interfaces*, 1997.

[17] William H. Bares and James C. Lester. Intelligent multi-shot visualization interfaces for dynamic 3d worlds. In *Intelligent User Interfaces*, pages 119–126, 1999.

[18] He Li-wei, Cohen Michael, and Salesin David. The virtual cinematographer: A paradigm for real-time camera control and directing. In *proceedings of SIGGRAPH*, 1996.

[19] Bill Tomlinson, Bruce Blumberg, and Delphine Nain. Expressive autonomous cinematography for interactive virtual environments. In Carles Sierra, Maria Gini, and Jeffrey S. Rosenschein, editors, *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 317–324, Barcelona, Catalonia, Spain, 2000. ACM Press.

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

[21] Mann W. and Thompson S. Rhetorical structure theory: A theory of text organization. *USC/ISI Technical Report*, 1987.

[22] Young R. M., Pollack M. E., and Moore J. D. Decomposition and causality in partial-order planning. In *proceedings of International Conference on Artificial Intelligence and Planning Systems*, Chicago, IL, 1994.

[23] M Maybury. Communicative acts for explanation generation. *International Journal for Man-Machine Studies*, 37:135–172, 1992.

[24] Charles Callaway, Elena Not, Alessandra Novello, Cesare Rocchi, Oliviero Stock, and Massimo Zancanaro. Automatic cinematography and multilingual nlg for generating video discourse. *To appear, Artificial Intelligence*, 2004.

[25] Charles B. Callaway. Narrative prose generation. *Artificial Intelligence*, 139:213–252, 2002.

[26] Marc Christie, Rumesh Machap, Jean-Marie Normand, Patrick Olivier, and Jonathan Pickering. Virtual camera planning: A survey. In *Smart Graphics*, pages 40–52, 2005.

[27] P. Karp and S. Feiner. Issues in the automated generation of animated presentations. In *Proceedings on Graphics interface '90*, pages 39–48, Toronto, Ont., Canada, Canada, 1990. Canadian Information Processing Society.

[28] Michael Gleicher and Andrew Witkin. Through-the-lens camera control. In *Computer Graphics*, pages 331–340, 1992.

[29] William H. Bares, Scott McDermott, Christina Boudreaux, and Somying Thainimit. Virtual 3d camera composition from frame constraints. In *ACM Multimedia*, pages 177–186, 2000.

[30] D. Amerson and S. Kime. Real-time cinematic camera control for interactive narratives, working notes of aaai spring symposium, spring,, 2000.

[31] Dan Amerson, Shaun Kime, and R. Michael Young. Real-time cinematic camera control for interactive narratives. In *Advances in Computer Entertainment Technology*, page 369, 2005.

[32] William Bares and James Lester. Cinematographic user models for automated realtime camera control in dynamic 3d environments. In *User Modeling: In Proceedings of the sixth international conference, UM97*, 1997.

[33] Jhala Arnav, Bares William, and R. Michael Young. Towards an intelligent storyboarding tool for 3d games. In *(to appear) ACM Advanced in Computer Entertainment 2005*, 2005.

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

[35] Michael Lebowitz. Creating a story-telling universe. In *IJCAI*, pages 63–65, 1983.

[36] Joseph Bates. Virtual reality, art and entertainment. *Presence*, 1(1):133–138, 1992.

[37] A. Bryan Loyall and Joseph Bates. Personality-rich believable agents thta use language. In *Agents*, pages 106–113, 1997.

[38] Marc Cavazza, Fred Charles, and Steven J. Mead. Character-based interactive storytelling. *IEEE Intelligent Systems*, 17(4):17–24, 2002.

[39] Michael Mateas and Andrew Stern. A behavior language for story-based believable agents. *IEEE Intelligent Systems*, 17(4):39–47, 2002.

[40] Earl David Sacerdoti. *A structure for plans and behavior.* PhD thesis, Stanford, CA, USA, 1975.

[41] Subbarao Kambhampati, Craig A. Knoblock, and Qiang Yang. Planning as refinement search: A unified framework for evaluating design tradeoffs in partial-order planning. *Artificial Intelligence*, 76(1-2):167–238, 1995.

[42] Richard Fikes and Nils J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artif. Intell.*, 2(3/4):189–208, 1971.

[43] Jhala Arnav. A cinematic camera planning system for dynamic narratives. *Master's thesis, NCSU*, 2004.

[44] Johanna D. Moore and Cecile Paris. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–694, 1994.

[45] E Hovy. Pragmatics and natural language generation. *Artificial Intelligence*, 43:153–197, 1990.

[46] Georges Polti. *Thirty-Six Dramatic Situations.* The Writer Inc., Boston, USA., 1921; 1977.

[47] Ronald Tobias. *Theme and Strategy.* Elements of Fiction Writing. Writer's Digest Books, Cincinnati, OH, 1990.

[48] R. Michael Young. Notes on using plan structures in the creation of interactive plot. In *The Working Notes of the AAAI Fall Symposium on Narrative Intelligence*, Cape Cod, MA, 1999. Association for the Advancement of Artificial Intelligence (AAAI), AAAI Press.

[49] R. Michael Young, Mark Riedl, Mark Branly, Arnav Jhala, R J Martin, and C J Saretto. An architecture for integrating plan-based behavior generation with interactive game environments, 2004.

[50] Jennifer Van Sijll. *Cinematic Storytelling.* Michael Wiese Productions, Studio City, CA, 2005.

[51] Mark Riedl and R. Michael Young. Character focused narrative planning. In *International Conference on Virtual Storytelling*, number Second, Toulouse, France, 2003.

[52] Ansen Dibell. *Plot.* Elements of Fiction Writing. Writer's Digest Books, Cincinnati, OH, 1988.

[53] William Noble. *Conflict, Action and Suspense.* Elements of Fiction Writing. Writer's Digest Books, Cincinnati, OH, 1994.

[54] S. Rimmon-Kenan. *Narrative Fiction: Contemporary Poetics.* Routledge, London, 1983.

[55] M. Ryan. *Possible Worlds, Artificial Intelligence and Narrative Theory.* Indiana University Press, Bloomington, IN, 1991.

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

[57] Neil Yorke-Smith, Kristen Brent Venable, and Francesca Rossi. Temporal reasoning with preferences and uncertainty. In *IJCAI*, pages 1385–, 2003.

[58] Thomas Vernieri. A web services approach to generating and using plans in configurable execution environments. Master's thesis, North Carolina State University, 2006.

[59] Arnav Jhala, Curtis Rawls, Samuel Munilla, and R. Michael Young. Longboard: A sketch based storyboarding tool for machinima. In *(21st Florida AI Research Society Conference (FLAIRS))*. AAAI, AAAI Press, November 2007.

[60] David Christian, David B. Christian, R. Michael Young, and R. Michael Young. Comparing cognitive and computational models of narrative structure. In *Proceedings of the 19 th National Conference on Artificial Intelligence*, pages 385–390, 2004.

[61] Arnav Jhala. Exploiting structure and conventions of movie scripts for data mining and information retrieval. In Ulrike Spierling and Nicolas Szilas, editors, *International Conference on Interactive Digital Storytelling*. LNCS.

[62] Byung-Chull Bae and R Michael Young. A use of flashback and foreshadowing for surprise arousal in narrative using a plan-based approach. In Ulrike Spierling and Nicolas Szilas, editors, *International Conference on Interactive Digital Storytelling*. LNCS.

[63] Yun-Gyung Cheong, Arnav Jhala, Byung-Chull Bae, and R. Michael Young. Automatically generating summary visualizations from game logs. In *AIIDE*, 2008.

Appendices

Appendix A

Dramatic Patterns Original 36 Dramatic Situation Patterns from Polti's book adapted and edited from the summarization by Trevor Lawrence. In this appendix, the patterns are labeled with their name, followed by the list of participants named by their role, then examples of different dramatic situations where the same pattern can be applied. In the formalism adopted in this dissertation, each pattern represents an abstract plan operator with variables representing participants in the story. Each situation example represents an expansion into specific action operators executed by participants in that situation.

I - Supplication

II - Deliverance

III - Vengeance of a crime

IV - Vengeance taken for kindred upon kindred

V - Pursuit

VI - Disaster

VII - Falling prey to cruelty or misfortune

VIII - Revolt

IX - Daring enterprise

X - Abduction

XI - The Enigma

XII - Obtaining

XIII - Enmity of kinsmen

XIV - Rivalry of kinsmen

XV - Murderous adultery

XVI - Madness

XVII - Fatal imprudence

XVIII - Involuntary crimes of love

XIX - Slaying of a kinsman unrecognised

XX - Self-sacrificing for an ideal

XXI - Self-sacrifice for kindred

XXII - All sacrificed for a passion

XXIII - Necessity of sacrificing loved ones

XXIV - Rivalry of superior and inferior

XXV - Adultery

XXVI - Crimes of love

XXVII - Discovery of the dishonour of a loved one

XXVIII - Obstacles to love

XXIX - An enemy loved

XXX - Ambition

XXXI - Conflict with a god

XXXII - Mistaken jealousy

XXIII - Erroneous judgement

XXXIV - Remorse

XXXV - Recovery of a lost one

XXXVI - Loss of loved ones

I - Supplication

”a persecutor, a suppliant and a power in authority”

A

1. fugitives imploring the powerful for help against their enemies
2. assistance implored for the performance of a pious duty which has been forbidden
3. appeals for a refuge in which to die

B

1. hospitality besought by the shipwrecked
2. charity entreated by those cast off by their own people, whom they have disgraced
3. expiation, the seeking of pardon, healing or deliverance
4. the surrender of a corpse, or of a relic, solicited

C

1. supplication of the powerful for those dear to the suppliant
2. supplication to a relative in behalf of another relative
3. supplication to a mother’s lover, in her behalf

II - Deliverance

”an unfortunate, a threatener, a rescuer”

A

1. appearance of a rescuer to the condemned B
1. a parent replaced upon the throne by his children
2. rescue by friends or by strangers grateful for benefits or hospitality

III - Vengeance of a crime

”an avenger and a criminal”

A

1. the avenging of a slain parent or ancestor
2. the avenging of a slain child or descendant
3. vengeance for a child dishonoured
4. the avenging of a slain wife or husband
5. vengeance for the dishonour or attempted dishonouring of a wife
6. vengeance for a mistress slain

7. vengeance for a slain or injured friend
8. vengeance for a sister seduced

B

1. vengeance for intentional injury or spoliation
2. vengeance for having been despoiled during absence
3. revenge for a false accusation
4. vengeance for violation
5. vengeance for having been robbed of one's own
6. vengeance on a whole sex for a deception by one

C

1. professional pursuit of criminals

IV - Vengeance taken for kindred upon kindred

"avenging kinsman; guilty kinsman; remembrance of the victim, a relative of both"

A

1. a father's death avenged upon a mother
2. a mother avenged upon a father

B

1. a brother's death avenged upon a son

C

1. a father's death avenged upon a husband

D

1. a husband's death avenged upon a father

V - Pursuit

"punishment and fugitive"

A

1. fugitives from justice pursued for brigandage, political offences, etc

B

1. pursued for a fault of love

C

1. a hero struggling against a power

D

1. a pseudo-madman struggling against an Iago-like alienist

VI - Disaster

”a vanquished power; a victorious enemy or a messenger”

A

1. defeat suffered
2. a fatherland destroyed
3. the fall of humanity
4. a natural catastrophe

B

1. a monarch overthrown

C

1. ingratitude suffered
2. the suffering of unjust punishment or enmity
3. an outrage suffered

D

1. abandonment by a lover or a husband
2. children lost by their parents

VII - Falling prey to cruelty or misfortune

”an unfortunate; a master or a misfortune”

A

1. the innocent made the victim of ambitious intrigue

B

1. the innocent despoiled by those who should protect

C

1. the powerful dispossessed or wretched
2. a favourite or an intimate finds himself forgotten

D

1. the unfortunate robbed of their hope

VIII - Revolt

"a tyrant and conspirator"

A

1. a conspiracy chiefly of one individual
2. a conspiracy of several

B

1. revolt of one individual who influences and involves others
2. a revolt of many

IX - Daring enterprise

"a bold leader; an object; an adversary"

A

1. preparations for war

B

1. war
2. a combat

C

1. carrying off a desired person or object
2. recapture of a desired object

D

1. adventurous expeditions
2. adventure undertaken for the purpose of obtaining a beloved woman

X - Abduction

"the abductor; the abducted; the guardian"

A

1. abduction of an unwilling woman

B

1. abduction of a consenting woman

C

1. recapture of the woman without the slaying of the abductor
2. recapture of the woman with the abductor slain

D

1. the rescue of a captive friend
2. the rescue of a child
3. the rescue of a soul in captivity to error

XI - The Enigma

”interrogator, seeker and problem”

A

1. search for person who must be found on pain of death

B

1. a riddle to be solved on pain of death
2. a riddle to be solved on pain of death in which the poser is the coveted woman

C

1. temptations offered with the object of discovering his name
2. temptations offered with the object of ascertaining the sex
3. tests for the purposes of ascertaining the mental condition

XII - Obtaining

”a solicitor and an adversary who is refusing, or an abitrator and opposing parties”

A

1. efforts to obtain an object by ruse or force

B

1. endeavour by means of persuasive eloquence alone

C

1. eloquence with an arbitrator

XIII - Enmity of kinsmen

”a malevolent kinsman; a hated or reciprocally hating kinsman”

A

1. one brother hated by several
2. reciprocal hatred between brothers
3. hatred between relatives for reasons of self-interest

B

1. hatred of father and son
2. mutual hatred
3. hatred of daughter for the father

C

1. hatred of grandfather for grandson

D

1. 1 hatred of father-in-law for son-in-law
2. hatred of two brothers-in-law

E

1. hatred of mother-in-law for daughter-in-law

F

1. infanticide

XIV - Rivalry of kinsmen

"the preferred kinsman; the rejected kinsman; the object"

A

1. malicious rivalry of a sibling
2. malicious rivalry of two siblings
3. rivalry of two siblings, with adultery on the part of one

B

1. rivalry of a parent and a child for an unmarried lover
2. rivalry of a parent and a child for a married lover
3. rivalry of a parent and a child for the love of the other parent
4. rivalry of parent and child

C

1. rivalry of cousins

D

1. rivalry of friends

XV - Murderous adultery

"two adulterers; a betrayed spouse"

A

1. slaying of a husband by, or for, a paramour
2. slaying of a trusted lover

B

1. slaying of a wife for a paramour, and in self-interest

XVI - Madness

”mad person and victim”

A

1. kinsmen slain in madness
2. a lover slain in madness
3. slaying or injuring of a person not hated

B

1. disgrace brought upon oneself through madness

C

1. loss of loved ones brought about by madness

D

1. madness brought on by fear of heredity insanity

XVII - Fatal imprudence

”the imprudent; the victim or the object lost”

A

1. imprudence the cause of one’s own misfortune
2. imprudence the cause of one’s own dishonour

B

1. curiosity the cause of one’s own misfortune
2. loss of possession of a loved one through curiosity

C

1. curiosity the cause of death or misfortune to others
2. imprudence the cause of a relative’s death
3. imprudence the cause of a lover’s death
4. credulity the cause of kinsmen’s deaths

5. credulity the cause of misfortune

XVIII - Involuntary crimes of love

”the lover; the beloved; the revealer”

A

1. discovery that one has married one’s mother
2. discovery that one has a sibling as a lover

B

1. discovery that one has married one’s sibling
2. discovery that one has married one’s sibling, in which the crime has been villainously planned by a third person
3. being on the point of taking a sibling, unknowingly, as a lover

C

1. being upon the point of violating, unknowingly, a child

D

1. being upon the point of committing an adultery unknowingly
2. adultery committed unknowingly

XIX - Slaying of a kinsman unrecognised

”the slayer; the unrecognised victim”

A

1. being upon the point of slaying a daughter unknowingly by command of a divinity or an oracle
2. being upon the point of slaying a daughter unknowingly through political necessity
3. being upon the point of slaying a daughter unknowingly through a rivalry in love
4. being upon the point of slaying a daughter unknowingly through hatred of the lover of the unrecognised daughter

B

1. being upon the point of slaying a son unknowingly
2. being upon the point of slaying a son unknowingly, strengthened by Machiavellian instigations
3. being upon the point of slaying a son unknowingly, strengthened by Machiavellian insti-

gations, intermixed with hatred of kinsmen

C

1. being upon the point of slaying a brother, unknowingly, in anger
2. a sister, upon the point of slaying a brother, unknowingly, through professional duty

D

1. slaying of a mother unrecognised

E

1. slaying of a father unknowingly, through Machiavellian advice
2. slaying of a father unknowingly
3. insulting of a father unknowingly
4. being on the point of slaying of a father unknowingly

F

1. a grandfather slain unknowingly, in vengeance and through instigation
2. a grandfather slain involuntarily
3. a father-in-law killed involuntarily

G

1. involuntary killing of a loved woman
2. being on the point of killing a lover, unrecognised
3. failure to rescue an unrecognised son

XX - Self-sacrificing for an ideal

"the hero; the Ideal' the "creditor" or the person or thing sacrificed"

A

1. sacrifice of life for the sake of one's word
2. life sacrificed for the benefit of one's own people
3. life sacrificed in filial piety
4. life sacrificed for the sake of one's faith

B

1. both love and life sacrificed for one's faith
2. both love and life sacrificed to a cause
3. love sacrificed to interests of state

C

1. sacrifice of well-being to duty

D

1. the ideal of ”honour” sacrificed to the ideal of ”faith”

XXI - Self-sacrifice for kindred

”the hero; the kinsman; the ”creditor” or person or thing sacrificed”

A

1. life sacrificed for that of a relative or loved one
2. life sacrificed for the happiness of a relative or loved one

B

1. ambition sacrificed for the happiness of a parent
2. ambition sacrificed for the life of a parent

C

1. love sacrificed for the sake of a parent’s life
2. love sacrificed for the happiness of one’s child
3. love sacrificed for the happiness of a loved one
4. love sacrificed for the happiness of one’s child, but the situation brought about by unjust laws

D

1. life and honour sacrificed for the life of a parent or loved one
2. modesty sacrificed for the life of a relative or a loved one

XXII - All sacrificed for a passion

”the lover; the object of the fatal passion; the person or thing sacrificed”

A

1. religious vows of chastity broken foir a passion
2. a vow of purity broken
3. a future ruined by a passion
4. power ruined by passion
5. ruin of mind, health and life
6. ruin of fortunes, lives and honours

B

1. temptations destroying the sense of duty, of pity, etc.

C

1. destruction of honour, fortune and life by erotic vice
2. destruction of honour, fortune and life by any other vice

XXIII - Necessity of sacrificing loved ones

”the hero; the beloved victim; the necessity for the sacrifice”

A

1. necessity for sacrificing a daughter in the public interest
2. duty of sacrificing a daughter in fulfilment of a vow to a god
3. duty of sacrificing benefactors or loved ones to one’s faith

B

1. duty of sacrificing one’s child, unknown to others, under the pressure of necessity
2. duty of sacrificing, unknown to others, one’s father, under the pressure of necessity
3. duty of sacrificing, unknown to others, one’s husband, under the pressure of necessity
4. duty of sacrificing a son-in-law for the public good
5. duty of sacrificing a son-in-law for the sake of reputation
6. duty of contending with a brother-in-law for the public good
7. duty of contending with a friend

XXIV - Rivalry of superior and inferior

”the superior rival; the inferior rival; the object of rivalry”

A

1. rivalry of a mortal and an immortal
2. rivalry of two divinities of unequal power
3. rivalry of a magician and an ordinary man
4. rivalry of conqueror and conquered
5. rivalry of victor and vanquished
6. rivalry of a master and a banished man
7. rivalry of suzerain king and vassal king
8. rivalry of a king and a noble
9. rivalry of a powerful person and an upstart

10. rivalry of rich and poor
11. rivalry of an honoured man and a suspected one
12. rivalry of two who are almost equal
13. rivalry of equals, one of whom has in the past been guilty of adultery
14. rivalry of a man who is loved and one who has not the right to love
15. rivalry of the two successive husbands of a divorcee

B

1. rivalry of a sorceress and an ordinary woman
2. rivalry of a victor and a prisoner
3. rivalry of a queen and a subject
4. rivalry of a queen and a slave
5. rivalry of a lady and a servant
6. rivalry of a lady and a woman of humbler position
7. rivalry of two who are almost equals, complicated by the abandonment of one
8. rivalry between a memory and an ideal (that of a superior woman) and a vassal of her own
9. rivalry of mortal and immortal

C

1. double rivalry (A vs B vs C vs D)

D

1. rivalry of two immortals
2. rivalry of two mortals
3. rivalry of two lawful wives

XXV - Adultery

"a deceived spouse; two adulterers"

A

1. a mistress betrayed for a young woman
2. a mistress betrayed for a young wife
3. a mistress betrayed for a girl

B

1. a wife betrayed for a slave, who does not love in return
2. a wife betrayed for debauchery
3. a wife betrayed for a married woman
4. a wife betrayed with the intention of bigamy
5. a wife betrayed for a young girl, who does not love in return
6. a wife envied by a young girl who is in love with her husband
7. a wife betrayed by a courtesan
8. rivalry between a wife who is antipathetic and a mistress who is congenial
9. rivalry between a generous wife and an impassioned girl

C

1. an antagonistic husband sacrificed for a congenial lover
2. a husband, believed to be lost, forgotten for a rival
3. a commonplace husband, sacrificed for a sympathetic lover
4. a good husband betrayed for an inferior rival
5. a good husband betrayed for a grotesque rival
6. a good husband betrayed for an odious rival
7. a good husband betrayed for a commonplace rival, by a perverse wife
8. a good husband betrayed for a rival less handsome, but useful

D

1. vengeance of a deceived husband
2. jealousy sacrificed for the sake of a cause

E

1. a husband persecuted by a rejected rival

XXVI - Crimes of love

”the lover; the beloved”

A

1. a mother in love with her son
2. a daughter in love with her father
3. violation of a daughter by a father

B

1. a woman enamoured of her stepson

2. a woman and her stepson enamoured of each other

3. a woman being the mistress, at the same time, of a father and son, both of whom accept the situation

C

1. a man becomes the lover of his sister-in-law
2. a man becomes enamoured of his sister-in-law
3. a brother and sister in love with each other

D

1. a man enamoured of another man, who yields

E

1. a woman enamoured of a bull

XXVII - Discovery of the dishonour of a loved one

"the dishonourer; the guilty one"

A

1. discovery of a parent's shame
2. discovery of a child's dishonour
3. discovery of a sibling's shame or dishonour

B

1. discovery of dishonour in the family of one's fiancee
2. discovery that one's wife has been violated before marriage
3. discovery that one's wife has been violated since the marriage
4. discovery that one's spouse has previously committed a fault
5. discovery that one's spouse has formerly been a prositute
6. discovery of dishonour on the part of a lover
7. discovery that one's mistress, formerly a prostitute, has returned to her former life
8. discovery that one's lover is of bad character
9. discovery that one's spouse is of bad character
10. 1discovery that one's lover is specifically weakened

C

1. discovery that one's son is an assassin

D

1. duty of punishing a traitorous relative
2. duty of punishing a son condemned under a law which the father has made
3. duty of punishing a son believed to be guilty
4. duty of sacrificing, to fulfil a vow of tyrranicide, a father until then unknown
5. duty of punishing a wrongdoing relative
6. duty of punishing one's mother to avenge one's father

XXVIII - Obstacles to love

"two lovers, an obstacle"

A

1. marriage prevented by inequality of rank
2. inequality of fortune an impediment to marriage

B

1. marriage prevented by enemies and contingent obstacles

C

1. marriage forbidden on account of the young woman's previous betrothal to another
2. marriage forbidden on account of the young woman's previous betrothal to another, complicated by an imaginary marriage of the beloved object

D

1. free union impeded by the opposition of relatives
2. family affection disturbed by the parents-in-law

E

1. free union impeded by the incompatibility of temper of the lovers

XXIX - An enemy loved

"the beloved enemy; the lover; the hater"

A

1. the loved one hated by kinsmen of the lover
2. the lover pursued by the brothers of his beloved
3. the lover hated by the family of his beloved
4. the lover is a son of a man hated by the kinsmen of his beloved
5. the lover is an enemy of the party of the woman who loves him

B

1. the lover is the slayer of the father of his beloved
2. the beloved is the slayer of the father of her lover
3. the beloved is the slayer of the brother of her lover
4. the beloved is the slayer of the husband of the woman who loves him, but who has previously sworn to avenge that husband
5. the beloved is the slayer of a previous lover of the woman who loves him, but who has previously sworn to avenge the dead lover
6. the beloved is a slayer of a kinsman of the woman who loves him
7. the beloved is the daughter of the slayer of her lover's father

XXX - Ambition

"an ambitious person; a thing coveted; an adversary"

A

1. ambition watched and guarded against by a kinsman or patriot friend
2. ambition watched and guarded against by a brother
3. ambition watched and guarded against by a relative or person under no obligation
4. ambition watched and guarded against by partisans

B

1. rebellious ambition

C

1. ambition and covetousness heaping crime upon crime
2. parricidal ambition

XXXI - Conflict with a god

"an immortal; a mortal"

A

1. struggle against a deity
2. strife with believers in a god

B

1. controversy with a deity
2. punishment for contempt of a god

3. punishment for pride before a god
4. presumptuous rivalry with a god
5. impudent rivalry with a deity

XXXII - Mistaken jealousy

"the jealous; the object of whose possessions he is jealous; the supposed accomplice; the cause or the author of the mistake"

A

1. the mistake originates in the suspicious mind of the jealous one
2. mistaken jealousy aroused by a fatal chance
3. mistaken jealousy of a love which is purely platonic
4. baseless jealousy aroused by malicious rumours

B

1. jealousy suggested by a traitor who is moved by hatred
2. jealousy suggested by a traitor who is moved by self-interest
3. jealousy suggested by a traitor who is moved by jealousy and self-interest

C

1. reciprocal jealousy suggested to husband and wife by a rival
2. jealousy suggested to the husband by a dismissed suitor
3. jealousy suggested to the husband by a woman who is in love with him
4. jealousy suggested to the wife by a scorned rival
5. jealousy suggested to a happy lover by the deceived husband

XXIII - Erroneous judgement

"the mistaken one; the victim of the mistake; the cause or author of the mistake; the guilty person"

A

1. suspicion where faith is necessary
2. false suspicion
3. false suspicions aroused by the a misunderstood attitude of a loved one
4. false suspicions aroused by indifference

B

1. false suspicions drawn upon oneself to save a friend
2. false suspicions fall upon the innocent
3. false suspicions fall upon the innocent spouse of the guilty one
4. false suspicions fall upon an innocent but guilty-intentioned
5. false suspicions fall upon an innocent who believes themselves guilty
6. a witness to a crime, in the interests of loved one, lets accusation fall upon the innocent

C

1. the accusation is allowed to fall upon an enemy
2. the error is provoked by the enemy
3. the mistake is directed against the victim by her brother

D

1. false suspicion thrown by the real culprit upon one of his enemies
2. false suspicion thrown by the real culprit upon the second victim against which he has plotted from the beginning
3. false suspicion thrown upon a rival
4. false suspicion thrown upon an innocent because he has refused to be an accomplice
5. false suspicion thrown by a deserted mistress upon a lover who left her because he would not deceive her husband
6. struggle to rehabilitate oneself and to avenge a judicial error purposely caused

XXXIV - Remorse

"the culprit; the victim or sin; the interrogator"

A

1. remorse for an unknown crime
2. remorse for a parricide
3. remorse for an assassination
4. remorse for the murder of husband or wife

B

1. remorse for a fault of love
2. remorse for an adultery

XXXV - Recovery of a lost one

”the seeker; the one found”

A

1. recovery of a lost one

XXXVI - Loss of loved ones

”a kinsman slain; a kinsman spectator; an executioner”

A

1. witnessing the slaying of kinsmen while powerless to prevent it
2. helping to bring misfortune upon one’s people through professional secrecy

B

1. divining the death of a loved one

C

1. learning of the death of a kinsman or ally

D

1. relapse into primitive baseness, through despair on learning of the death of a loved one

Appendix B

Classification of Camera Shots

A classification of different cinematic techniques according to different properties of the story world. These properties can be represented by operators in the Darshak system. Examples of how these rules are implemented are showin in Appendix **??**. For instance, camera motion actions like tracking are implemented as primitive operators and composition actions like directing the eye and orientation are implemented as variations on the Look-At operators. All of these are listed for completeness, the techniques that have corresponding implemented operators are marked with an asterix (*). Some of these, like the depth of field, are not implemented in the system due to the limiations of the game engine in producing the effects. Character dialouge is implemented as textual sub-titles.

Space

- One, two and three dimensions

- Lines and Planes and Action *
- Depth of field (Z- Axis)
- Shifting perspective

Composition

- Directing the eye *
- Balance *
- Imbalance
- Orientation *
- Size/Emphasis *

Blocking

- Circular
- Linear
- Triangular
- Rectangular
- Organic/Geometric

Editing

- Montage
- Assembly
- Mise-en-scene
- Intercutting *
- Split-Screen
- Dissolve
- Smash Cut

Time

- Pacing *

  - Contrast of Time
  - Expanding Time
  - Slow-Motion
  - Fast-Motion
  - Flashback
  - Flashforward
  - Freeze Frame
  - Foreshadowing

Sound

  - Character Dialogue *
  - Diegetic Sound (Ambience)
  - Extra-Diegetic (Expressive Sound)
  - Surreal Sound (Meta-Diegetic)

Transitions

  - Match Cut (Spatial)
  - Match Cut (Temporal)
  - Match Cut (Action) *
  - Match Cut (Idea)
  - Match Dissolve (Time Transition)
  - Disrupted Match Cut
  - Audio Matching

Lens

  - Wide Angle
  - Telephoto
  - Fisheye

Angles

- High Angle *
- Low Angle *

Motion

- Track *
- Pan *
- Dolly
- Zoom *

Appendix C

Operator Library

Operators from Darshak's operator library for the WestWorld Lane domain used in the experiments. The key elements of the operators are shown. All operators have temporal annotations and constraints on temporal variables as shown in the first few operators. These constraints are shown in operators where they differ from their usual usage as shown in operators in the beginning. Additional steps, like the start and end step, are added by the planner when abstract operators are expanded. These are not explicitly shown unless they differ from the standard operator definition. Detailed operator representation in Darshak, with an explanatin of the each part, is given in Section 3.2.

Primitive Operators

Operator

Type : LookAt

Parameters : ?f, ?shot-type, ?dir ?Tend ?Tstart

Preconditions : none

Constraints : (>Tend Tstart) (shot-type ?shot-type) (direction ?dir) (time ?Tend) (time ?Tstart)

Effects : (infocus ?f)@[Tstart, Tend)

Operator

Type : LookAt-Close

Parameters : ?f, ?shot-type, ?dir ?Tend ?Tstart

Preconditions : none

Constraints : (>Tend Tstart) (shot-type shot-close) ...

Effects : (infocus ?f)@[Tstart, Tend)

Operator

Type : LookAt-medium

Parameters : ?f, ?shot-type, ?dir ?Tend ?Tstart

Preconditions : none

Constraints : (>Tend Tstart) (shot-type shot-medium) ...

Effects : (infocus ?f)@[Tstart, Tend)

Operator

Type : LookAt-long

Parameters : ?f, ?shot-type, ?dir ?Tend ?Tstart

Preconditions : none

Constraints : (>Tend Tstart) (shot-type shot-long) ...

Effects : (infocus ?f)@[Tstart, Tend)

Operator

Type : LookAt-front

Parameters : ?f, ?shot-type, ?dir ?Tend ?Tstart

Preconditions : none

Constraints : (>Tend Tstart) (dir front) ...

Effects : (infocus ?f)@[Tstart, Tend)

Operator

Type : LookAt-left

Parameters : ?f, ?shot-type, ?dir ?revdir ?Tend ?Tstart

Preconditions : none

Constraints : (>Tend Tstart) (dir left) ...

Effects : (infocus ?f)@[Tstart, Tend)

Operator

Type : Track

Parameters : ?f, ?shot-type, ?dir ?revdir

Preconditions :

(infocus ?f)@[Tstart)

(not (tracking ?f ?shot-type ?revdir)@[T1, Tstart))

Constraints : (>Tend Tstart)...

Effects :

(infocus ?f)@[Tstart, Tend)

(tracking ?f ?shot-type ?dir)@[Tstart, Tend)

Operator

Type : Track-with-actor

Parameters : ?f, ?shot-type, ?dir ?revdir ?Tstart ?Tend

Preconditions :

(infocus ?f)@[Tstart)

(not (tracking ?f ?shot-type ?revdir)@[T1, Tstart))

Constraints : (>Tend Tstart) (actor ?f) ...

Effects :

(infocus ?f)@[Tstart, Tend)

(tracking ?f ?shot-type ?dir)@[Tstart, Tend)

Operator

Type : Pan

Parameters : ?f, ?shot-type, ?dir ?revdir ?Tstart ?Tend

Preconditions :

(infocus ?f)@[Tstart)

(not (panning ?f ?shot-type ?revdir)@[T1, Tstart))

Constraints : (>Tend Tstart)...

Effects :

(infocus ?f)@[Tstart, Tend)

(tracking ?f ?shot-type ?dir)@[Tstart, Tend)

Operator

Type : Track-with-actor

Parameters : ?f, ?shot-type, ?dir ?revdir ?Tstart ?Tend

Preconditions :

(infocus ?f)@[Tstart)

(not (tracking ?f ?shot-type ?revdir)@[T1, Tstart))

Constraints : (>Tend Tstart) (actor ?f) ...

Effects :

(infocus ?f)@[Tstart, Tend)

(tracking ?f ?shot-type ?dir)@[Tstart, Tend)

Abstract Operators: Cinematic Idioms and Narrative Patterns

Show-Dialog (?speaker ?hearer ?prop ?act)@[?ts, ?te)
Description:
Constraints:
    (actor ?speaker) (actor ?hearer) (step ?act)
    (start ?act ?ts) (end ?act ?te)
Preconditions:
Effects:
    (Bel V (Speaks ?character ?prop)
Steps:
    (Step1 (Look-At ?character)@[ts, te))

Show-Reaction (?speaker ?hearer ?prop ?act)@[ts, te)
Description:
Constraints:
    (actor ?speaker) (actor ?hearer) (step ?act)
    (start ?act ?ts)(end ?act ?te)
Preconditions:
Effects:
    (Bel V (Bel ?hearer ?prop)))
Steps:
    (Step1 (Look-At ?hearer)@[?ts, ?te))

Show-Dialog2 (?speaker ?hearer ?prop ?act)@[?ts, ?te)
Description:
Constraints:
(actor ?speaker) (actor ?hearer) (step ?act)
(start ?act ?ts)(end ?act ?te)
Preconditions:
Effects:
(Bel V (Speaks ?character ?prop))
(Bel V (Bel ?hearer ?prop)))
Steps:
(Step1 (Look-At ?speaker)@[?ts, ?ts+2))
(Step2 (Look-At ?hearer))@[?ts+2, ?ts+4))
(Step3 (Look-At ?speaker))@[?ts+4, ?te))

Show-TwoConversation (?conv ?steplist)@[?ts, ?te)
Description:
Constraints:
(conversation ?conv) (steps ?conv ?steplist)
(mainparticipant ?conv ?p1)(secondparticipant ?conv ?p2)
(start ?act ?ts)(end ?act ?te)
Preconditions:
Effects:
(Bel V (Occurs (Conversation ?steplist)))
Steps:
(Step1 (Two-Shot ?p1 ?p2))
(Step2 (forall ?step in ?steplist)
(Show-Dialog ?step ?char)

Show-Action (?act ?actor)@[?ts, ?te)
Description:
Constraints:
(step ?act) (agent ?act ?actor) (effect ?act ?e)
(start ?act ?ts)(end ?act ?te)
Preconditions:
Effects:
(Bel V (Occurs ?act))
(Bel V (Agent ?act ?actor)))
(Bel V ?e))
Steps:
(Step1 (LookAt-long ?actor))

Show-Action-Dramatic (?act ?actor)@[?ts, ?te)
Description:
Constraints:
(step ?act) (agent ?act ?actor) (effect ?act ?e)
(start ?act ?ts)(end ?act ?te)
Preconditions:
Effects:
(Bel V (Occurs ?act)) (Bel V ?e))
(Bel V (Agent ?act ?actor)))
Steps:
(Step1 (LookAt-long ?actor))
(Step2 (LookAt-medium ?actor))
(Step3 (LookAt-close ?actor))
(Step4 (LookAt-medium ?actor))

Show-Move-Action (?act ?actor)@[?ts, ?te)
Description:
Constraints:
(step ?act) (agent ?act ?actor) (act-type ?act MOVE)
(start ?act ?ts)(end ?act ?te)
Preconditions:
Effects:
(Bel V (Occurs ?act))
(Bel V (Agent ?act ?actor)))
Steps:
(Step1 (TrackActor ?actor))

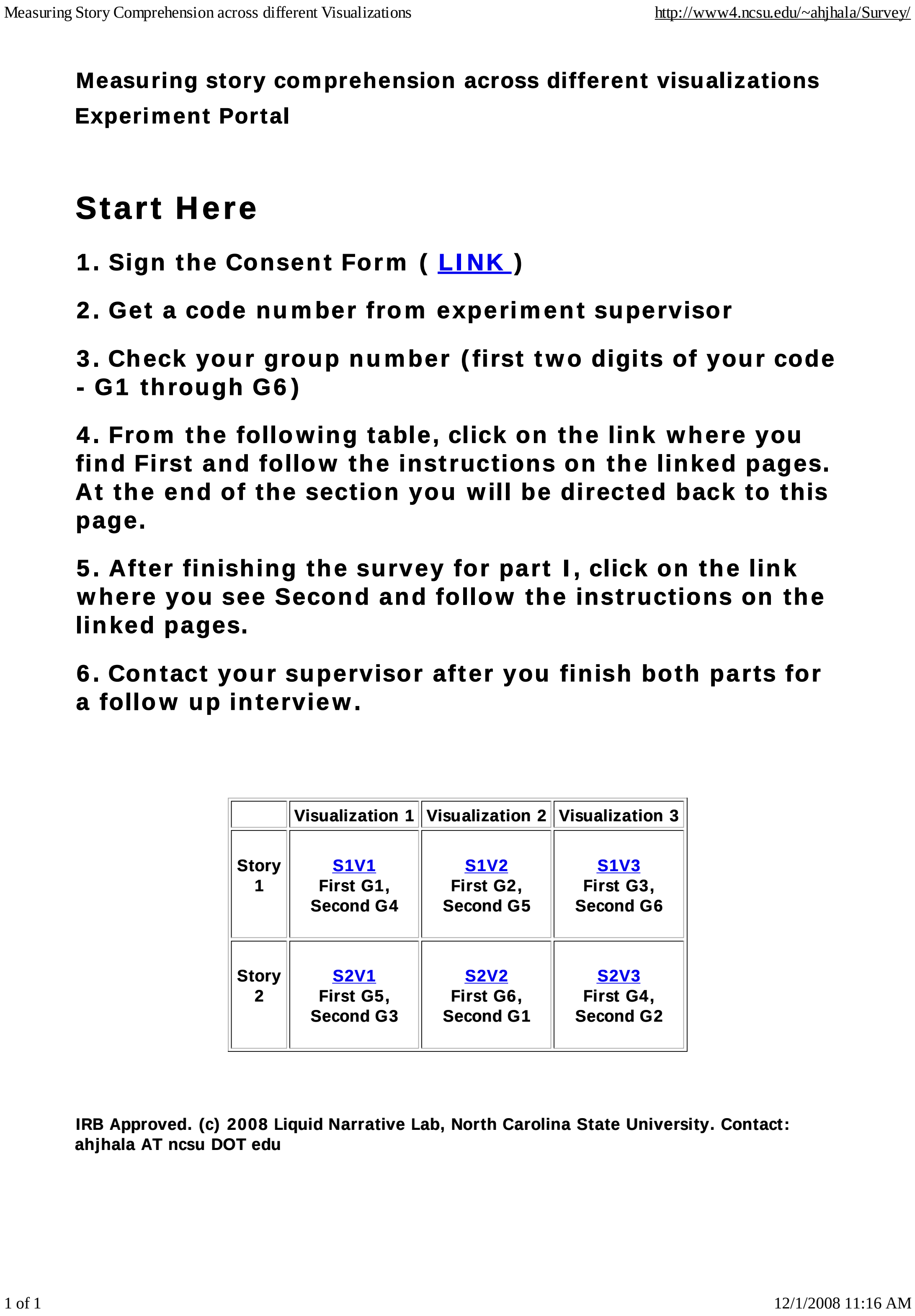
Show-Gunfight (?prot ?vil)@[?ts, ?te)
Description:
Constraints:
(actor ?prot) (actor ?vil)
(start ?act ?ts)(end ?act ?te)
Preconditions:
Effects:
(Bel V (Speaks ?character ?prop))
(Bel V (Bel ?hearer ?prop)))
Steps:
(Step1 (Two-Shot ?prot ?vil))
(Step2 (LookAt-medium ?prot))
(Step3 (LookAt-long ?vil))
(Step4 (Two-Shot ?prot ?vil))

Show-Robbery-Action (?act ?actor)@[?ts, ?te)
Description:
Constraints:
(step ?act) (agent ?act ?actor) (effect ?act ?e)
(start ?act ?ts)(end ?act ?te)
Preconditions:
Effects:
(Bel V ?e))
Steps:
(Step1 (Show-Action-Dramatic ?act ?actor))

Robbery (?character ?thing)@[?ts, ?te)
Description:
Constraints:
(actor ?character) (object ?thing)
(storyaction ?robbingact) (effect ?robbingact (has ?character ?thing))
(causallink ?enablers ?robbingact)
(start ?act ?ts)(end ?act ?te)
Preconditions:
(**Effects**:
(Bel V (has ?character ?thing)))
Steps:
(forall ?enabler in ?enablers
(Step1 (Show-Action ?enabler ?character)))
(Step2 (Show-Robbery-Action ?robbingact ?character)

APPENDIX D

Experiment Materials and Ouptut Screenshots This appendix lists the experiment materials used during evaluation. The experiment portal, instructions, consent form, first pages of survey questions, and some still frames from the stories are listed.



Measuring Story Comprehension across different Visualizations http://www4.ncsu.edu/~ahjhala/Survey/

Measuring story comprehension across different visualizations
Experiment Portal

Start Here

1. Sign the Consent Form (LINK)

2. Get a code number from experiment supervisor

3. Check your group number (first two digits of your code - G1 through G6)

4. From the following table, click on the link where you find First and follow the instructions on the linked pages. At the end of the section you will be directed back to this page.

5. After finishing the survey for part I, click on the link where you see Second and follow the instructions on the linked pages.

6. Contact your supervisor after you finish both parts for a follow up interview.

	Visualization 1	Visualization 2	Visualization 3
Story 1	S1V1 First G1, Second G4	S1V2 First G2, Second G5	S1V3 First G3, Second G6
Story 2	S2V1 First G5, Second G3	S2V2 First G6, Second G1	S2V3 First G4, Second G2

IRB Approved. (c) 2008 Liquid Narrative Lab, North Carolina State University. Contact: ahjhala AT ncsu DOT edu

1 of 1 12/1/2008 11:16 AM

Measuring story comprehension

Informed Consent Form

TITLE OF RESEARCH: Measuring the effects of different visualization techniques on story understanding.

INVESTIGATOR: Arnav Jhala, Dr. R Michael Young

SPONSOR: Self

Explanation of Procedures

We are asking you to take part in a research study. This research study is being conducted to measure effects of different visualization techniques on story understanding. If you enter the study, you will be shown two 4-minute short story videos and will be asked to fill a questionnaire with questions and answers about the story. You will be asked to rate the quality of each answer for all questions. The questionnaire will be provided to you on a computer and you will be asked to select your choices on a web based form. The total time for the study will be 1 Hour.

Risks and Discomforts

None.

Benefits

You may not benefit directly from taking part in this study. However, this study may help us better understand the effects of different visualizations on story understanding.

Confidentiality

Any information we collect from you during the experiment will be kept confidential.

Measuring Story Comprehension Across Different Visualizations

Instructions

You are asked to watch a short story video. Afterward, you will be presented with a series of questions about the story. An answer will be given for each question. You will be asked to rate whether the given answer is a good answer to that question or a bad answer to that question by selecting one of four options: very bad answer, somewhat bad answer, somewhat good answer, or very good answer. For example, read the story below.

There is a boy named Jack. Jack ran out of water at the house. Jack went up the hill. Jack fetched a pail of water. Jack fell down and broke his crown.

Questions and corresponding answers will be provided, such as the following. You do not need to rate the example questions.

Q: Why did Jack go up the hill?
A: Because Jack wanted to fetch a pail of water.

Very Bad Answer	*Somewhat Bad Answer*	*Somewhat Good Answer*	*Very Good Answer*

Q: Why did Jack go up the hill?
A: Because Jack ran out of water at the house.

Very Bad Answer	*Somewhat Bad Answer*	*Somewhat Good Answer*	*Very Good Answer*

Q: Why did Jack go up the hill?
A: Because Jack wanted to fall down and break his crown.

Very Bad Answer	*Somewhat Bad Answer*	*Somewhat Good Answer*	*Very Good Answer*

You do not need to answer the questions, merely select the rating that most appropriately describes the quality of the answer for each question. You may see the same question several times, but each time it will be paired with a slightly different answer. There is no right or wrong answer and you are not being evaluated. You may wish to watch the video several times before you begin, although you cannot refer back to the story as often as you wish. You are not being timed so take as much time as you need. Here are the characters that you will see in the video.


Lazarus Lane


Sheriff Bob


Outlaw Vinny


Bartender Bart


Teller Elsie

Click Here to Begin Experiment

Measuring Story Comprehension Across Different Visualizations

Instructions

You are asked to watch a short story video. Afterward, you will be presented with a series of questions about the story. An answer will be given for each question. You will be asked to rate whether the given answer is a good answer to that question or a bad answer to that question by selecting one of four options: very bad answer, somewhat bad answer, somewhat good answer, or very good answer. For example, read the story below.

There is a boy named Jack. Jack ran out of water at the house. Jack went up the hill. Jack fetched a pail of water. Jack fell down and broke his crown.
Questions and corresponding answers will be provided, such as the following. You do not need to rate the example questions.

Q: Why did Jack go up the hill?
A: Because Jack wanted to fetch a pail of water.

Very Bad Answer	*Somewhat Bad Answer*	*Somewhat Good Answer*	*Very Good Answer*

Q: Why did Jack go up the hill?
A: Because Jack ran out of water at the house.

Very Bad Answer	*Somewhat Bad Answer*	*Somewhat Good Answer*	*Very Good Answer*

Q: Why did Jack go up the hill?
A: Because Jack wanted to fall down and break his crown.

Very Bad Answer	*Somewhat Bad Answer*	*Somewhat Good Answer*	*Very Good Answer*

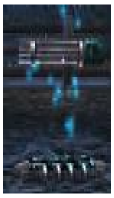
You do not need to answer the questions, merely select the rating that most appropriately describes the quality of the answer for each question. You may see the same question several times, but each time it will be paired with a slightly different answer. There is no right or wrong answer and you are not being evaluated. You may wish to watch the video several times before you begin, although you cannot refer back to the story as often as you wish. You are not being timed so take as much time as you need. Here are the characters that you will see in the video.

Lane

Billy

Blacksmith

Padre



Gatling Gun

Click Here to Begin Experiment

Measuring story comprehension Survey

Enter your user code: ☐

Q: Why did Lane challenge Vinny?
A: Because Lane heard that Vinny killed deputy sheriff.

Select the quality of the answer from the following:

○ **Very Bad Answer** ○ **Somewhat Bad Answer** ○ **Somewhat Good Answer** ○ **Very Good Answer**

Q: How did Lane become Deputy Sheriff?
A: By killing Vinny.

Select the quality of the answer from the following:

○ **Very Bad Answer** ○ **Somewhat Bad Answer** ○ **Somewhat Good Answer** ○ **Very Good Answer**

Q: How did Lane become Deputy Sheriff?
A: By winning Sheriff Bob's favor.

Select the quality of the answer from the following:

○ **Very Bad Answer** ○ **Somewhat Bad Answer** ○ **Somewhat Good Answer** ○ **Very Good Answer**

Q: Why did Lane challenge Vinny ?
A: Because Lane wanted to become Sheriff's Deputy.

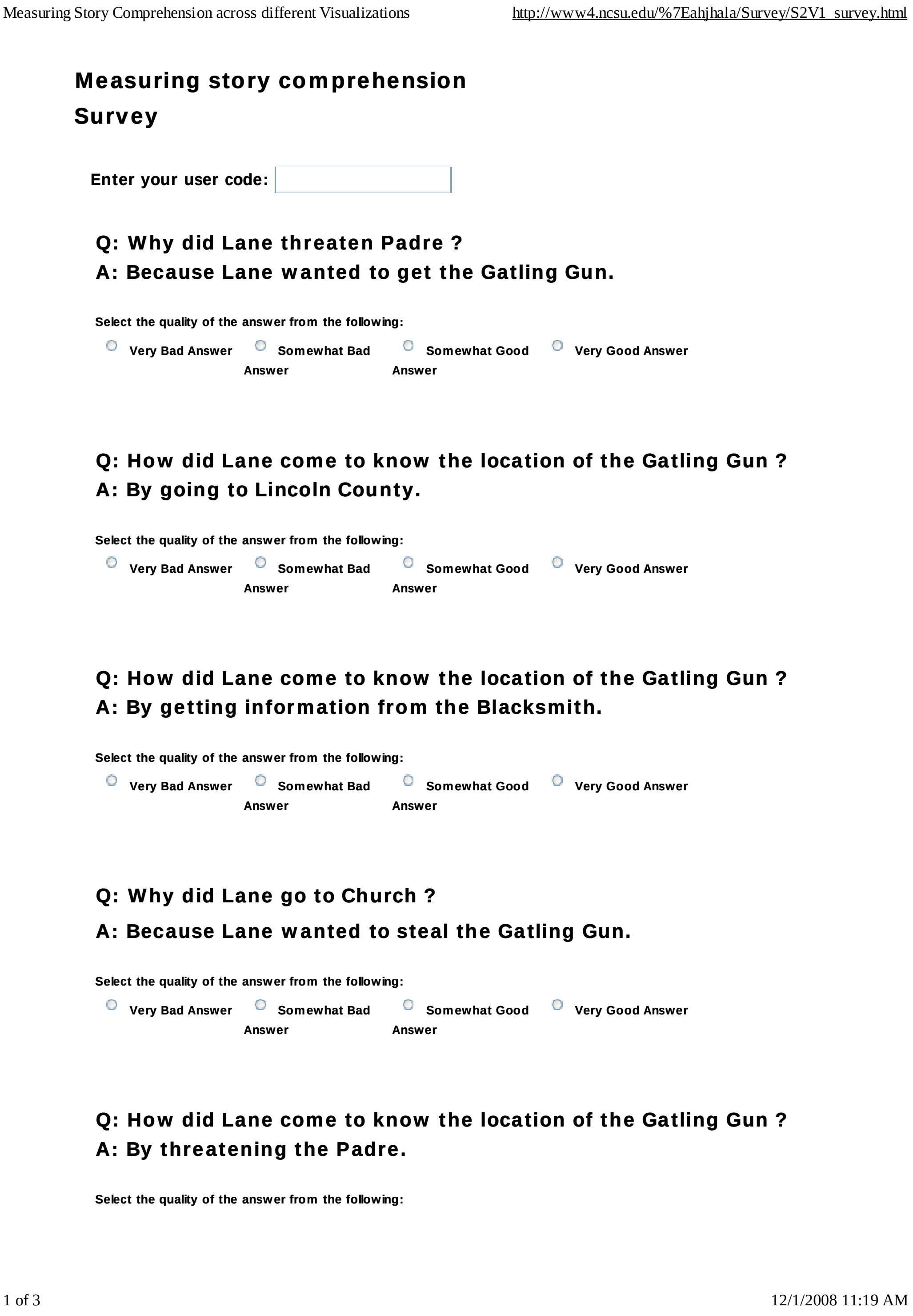
Select the quality of the answer from the following:

○ **Very Bad Answer** ○ **Somewhat Bad Answer** ○ **Somewhat Good Answer** ○ **Very Good Answer**

Q: What enabled Lane stealing tax money ?
A: Teller being threatened by him.

Select the quality of the answer from the following:

○ **Very Bad Answer** ○ **Somewhat Bad Answer** ○ **Somewhat Good Answer** ○ **Very Good Answer**



Measuring story comprehension Survey

Enter your user code: ☐

Q: Why did Lane threaten Padre ?
A: Because Lane wanted to get the Gatling Gun.

Select the quality of the answer from the following:
○ Very Bad Answer ○ Somewhat Bad Answer ○ Somewhat Good Answer ○ Very Good Answer

Q: How did Lane come to know the location of the Gatling Gun ?
A: By going to Lincoln County.

Select the quality of the answer from the following:
○ Very Bad Answer ○ Somewhat Bad Answer ○ Somewhat Good Answer ○ Very Good Answer

Q: How did Lane come to know the location of the Gatling Gun ?
A: By getting information from the Blacksmith.

Select the quality of the answer from the following:
○ Very Bad Answer ○ Somewhat Bad Answer ○ Somewhat Good Answer ○ Very Good Answer

Q: Why did Lane go to Church ?

A: Because Lane wanted to steal the Gatling Gun.

Select the quality of the answer from the following:
○ Very Bad Answer ○ Somewhat Bad Answer ○ Somewhat Good Answer ○ Very Good Answer

Q: How did Lane come to know the location of the Gatling Gun ?
A: By threatening the Padre.

Select the quality of the answer from the following:


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.