ABSTRACT

JHALA, ARNAV HARISH. An Intelligent Cinematic Camera Planning System for Dynamic Narratives. (Under the direction of R Michael Young).

This thesis presents a framework for automatic generation of cinematic discourse of a dynamic story. The motivation of this research is provided by the need for more expressive camera control in the current dynamic story generation systems and the lack of formal research in the area of visual discourse processing/generation systems.

Film directors and cinematographers have developed effective visual storytelling techniques. They have also articulated various rules for conveying the story to the viewer. The stereotypical ways of filming shot sequences are termed as *Idioms*. This thesis begins to formalize film idioms as plan operators, augmented with the *intentional* goals of the director, that represent *communicative acts* analogus to speech acts used in traditional natural language discourse planning systems [18,26,22]. Discourse processing/planning systems have focused on generation of natural language discourse. This is an attempt to extend this research to the generation of visual discourse. The main questions that are addressed in this thesis are:

- How does a visual communicative act change the model of the viewer and how can this be encoded in a formalism?
- How can the presentation of a scene/shot relate to the actions taking place in the story world and the information being conveyed to the viewer about the story world?

- What are the syntax and semantics of the visual medium of communication as they are specified by legal plan structures?

I present a study of film idioms and their formalization as plan operators followed by a formal description of the viewer model used by the system. Next, I discuss the representation of the story world plan that is communicated to the viewer by the discourse planner and a formal definition of the extended discourse planning algorithm. Finally, a sample scenario for the communicative plan generated by the discourse planner written in Lisp and executed on the Unreal Tournament 2003(UT) game engine used as a visualizer.

AN INTELLIGENT CINEMATIC CAMERA PLANNING SYSTEM FOR DYNAMIC NARRATIVES

by

JHALA ARNAV HARISH

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

COMPUTER SCIENCE

Raleigh

2004

APPROVED BY:

Dr. James Lester	Dr. Robert StAmant

Dr. R Michael Young

BIOGRAPHY

Arnav Jhala is a graduate student in the computer science department at NC State University. He earned his bachelors degree in computer engineering from Gujarat University in 2001. As a graduate student, Arnav has focused his studies on artificial intelligence and virtual worlds. His future interests lie in application of AI techniques to interactive entertainment applications like movies and video games.

ACKNOWLEDGEMENTS

I would like to thank, first of all my parents Harish and Kapila Jhala, for their immense support, advice and unconditional love. Without their support and encouragement, I would not even be here. I am also deeply thankful to my advisor, Dr. R Michael Young, for giving me this opportunity to work on an interesting topic, and patiently guiding me through my first hesitant steps in doing research.

Working in the Liquid Narrative lab with all the mimites is always fun and has made difficult times easier to survive. Mark Riedl and David Christian have been very helpful, especially in helping me deal with my initial hiccups learning Lisp and programming on longbow. I appreciate the long and insightful discussions on this work with Dr. James Lester and Dr. William Bares. The knowledge obtained through the human-computer interaction course taught by Dr Robert St. Amant has been invaluable for application to the interaction issues in the implementation of this system.

A very special mention goes to my friends and roommates Sandeep, Ashish, Kuldip, Ashwin, Kunal, Nirmit, Ajay, Sameer and Kurang for their understanding and support and for bearing with my weirdness through grad school.

Finally, to my whole family and closest friends who always have been there and who have sculpted my personality through childhood and youth.

TABLE OF CONTENTS

List of Figures

List of Tables

1. INTRODUCTION

A picture is worth a thousand words.
Anonymous

Cognitive Psychologists define narrative as one of the fundamental ways we organize information. The narrative theorist Chatman[7] defines narrative as being composed of two elements – story and discourse. The story part of a narrative specifies the abstract plot structure leading to the realization of goal(s) of the narrative. The discourse part of the narrative is the telling of a story and deals with how the story is communicated to the target audience. Narrative discourse can occur within different media, as illustrated in Figure 1.

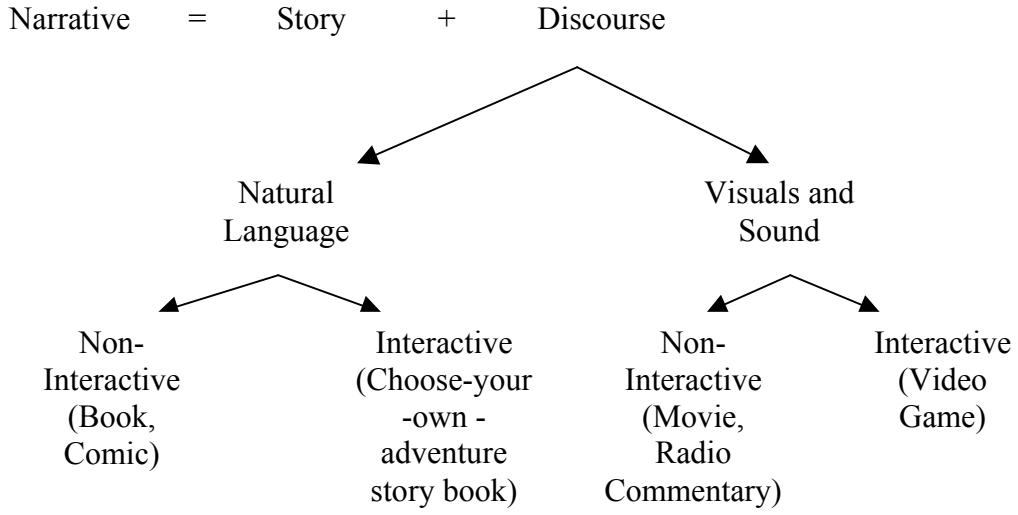

Figure 1 Structure of a Narrative

The field of cinema has developed effective techniques for generating discourse in a visual medium(i.e. movies) through the use of the motion picture camera. While the techniques of composing the camera for cinematic presentation of actions occuring in the real world are hand-crafted by cinematographers, there is a need in virtual 3D environments for a program that automatically controls the camera in order to realize similar storytelling capabilities.

In this thesis, I establish a model for intelligent generation of cinematic discourse of a story through a visual medium(i.e. 3D graphical virtual worlds). This goal is achieved by

generating specifications for controlling a virtual camera using techniques adapted from approaches to the generation of natural language discourse.

In the field of computer graphics, computer animation and games researchers (Drucker[10], Christianson[9], Bares[4]) have been concerned about the camera placement problem, that is, the need to develop a system that decides *how* to move the camera in a virtual world, finding the best possible position and movement of the camera given the geometric constraints of a story's virtual set. Some researchers[29] have also addressed the question of *why* a camera should move in a 3D world, that is, how a system could represent the cinematic motivation behind imposition of constraints for the camera to find better overall solutions for the problem of camera placement.

Early approaches to automatic camera control were restricted to optimizing geometric camera placement for individual frames. Some systems like [9] have incorporated cinematic rules and idioms in their implementation; their main focus has been the automatic specification of smooth geometric transitions for the camera. Because of their focus on localized transisions of shots, these approaches to camera planning have not attempted to capture the coherence of a sequence of shots with respect to the rhetorical content that is being conveyed to the viewer. In order to generate a mentally coherent narrative discourse, a camera planner should have a representation of the underlying plot and the causal as well as temporal relationships of actions and events occurring in the world. While work in automatic camera control has not addressed the creation of coherent narrative discourse, this problem(or its text-based analog) has been at the center of much work in the field of computational linguistics.

Within the computational linguistics community, research in the domain of discourse processing has explored both how and why speakers produce Illocutionary acts in order to change the beliefs of other agents. Grosz and Sidner's theory of discourse structure [15], for instance, considers a model of discourse containing elements of attention, intention and beliefs of the discourse participants used to generate and/or interpret coherent text discourse segments. This model and others similar to it (e.g., Rhetorical Structure Theory[23]) have been used in discourse planners [18,22,26], programs that

generate discourse by formalizing communicative acts as plan operators and use goal oriented reasoning to produce multi-sentential discourse structure.

A number of film theorists (Monaco [25], Arijon [3], Mascelli [21]) have studied films as a medium of communication similar to natural language. This thesis defines the problem of camera placement as a problem of generating communicative acts by providing visual information in place of speech acts considered by conventional discourse generation systems. In our work, a narrative planner is used to create a plan data structure as a source for a story representation by the intentional structure of the discourse plan. In this approach, primitive camera actions (e.g. look-at, track, pan, dolly) and cinematic idioms (stereotypical ways of filming certain sequences of shots adopted by cinematographers) are formalized as plan operators. The high-level communicative intentions of the cinematographer are represented as goals provided to the camera planner. The camera planner then produces a sequence of shots structured effectively convey the intended aspects of the story being filmed. To provide an architectural framework for the overall process, I have modeled the film production hierarchy developed and used by cinematographers in the design of the FILM(Film Idiom Language and Model) system, an implementation of the theoretical ideas described in the following sections of this thesis.

The contributions of this thesis are:

- The application of conventional discourse models to a new mode of communication, through a virtual camera in a 3D environment, as communicative actions that convey information about an unfolding storyline to the viewer.
- The development of a model that paralleles film production process in a computational framework, with a formal representation for shots and sequences, for capturing the cinematic effect in the form of intentions of the cinematographer.
- The formalization of film idioms and camera primitives as communicative plan operators for a decompositional partial order causal link planner with temporal and spatial constraints.

1.1 Organization of the following chapters

A brief study of the aspects of film theory that are relevant to this work is presented in section 2.1 followed by a discussion of the theory of discourse structure and its role in discourse planning systems previously used to generate multi-sentential textual discourse. The implementation of the FILM(Film Idiom Language and Model) system, a system that uses shot descriptions generated by a partial order planning algorithm to drive the camera within a 3D game world, is discussed in Chapter 3. A formal description of the camera-planning algorithm and example of its use is presented in chapter 4, followed by conclusions and discussion about future work on the system.

2. Background

It is a capital mistake to theorize before one has data.
-Arthur Conan Doyle

2.1 FILM THEORY

Film theory is the specification of concise, systematic concepts that apply to production of film and video. In this section I present the main concepts from film theory that are relevant to this work.

2.1.1 The Language of Film

Drawing of pictures was among the first forms of communication developed by humans. Although symbolic languages have dominated human communication, with the development of technology, movies are reaching a wider audience than books. The main differences between spoken/written language and the language of film or pictures can be noted as follows:

The way a written or spoken sentence in natural language is interpreted by a reader/hearer is different from the way a visual is interpreted by a viewer. Natural language discourse leaves the visualization to the reader's imagination, while visuals explicitly present the visualization to the viewer.

A film cannot be broken down into well defined syntactic units like words and sentences. This makes it harder to compare or represent the amount of information conveyed by a sentence in natural language compared to that contained in a clip from a movie.

Words in natural language have meanings associated with them. The specific meaning associated with each word is called the *Denotative* meaning of the word. The same word, when spoken in different contexts, conveys different meanings, these are called *Connotative* meanings. The well-defined syntax and semantics of natural language make it relatively easier to identify the connotative meanings of words. In film, creating sequences of shots that conveys the connotative meaning of a particular shot, is the art of

the cinematographer. The syntax and semantics of how to create such sequences of shots are not well-defined.

A very detailed discussion of film as a language is presented by Monaco [25]. Although due to the differences between film and language described above, it is hard to establish strict linguistic rules on films, they do convey meaning like conventional linguistic structures. The meaning that is conveyed by a sequence of shots can be considered analogous to that conveyed by a paragraph (or a sequence of sentences) in a book. This is our motivation behind using the discourse generation techniques to create movies for narrative planners that use the same representation. Films have proved to be a succesful medium of communication, and modeling the film hierarchy gives us the tools to develop systems where the main mode of communication is through visuals and sound. Our artificial intelligence system models the knowledge and reasoning of a cinematographer or a director to produce sequences of shots. One direct application of such a tool is in computer games, where there is a narrative structure that leads a user to accomplish certain tasks, to assist the user in focusing on the tasks that would lead to a successful solution. Another application is for a 3D motion storyboarding tool that automatically sets up standard shot blocking and gives suggestions to directors based on the input script.

2.1.2 Components of Film production process

Here I present a brief introduction to the process of film production. Figure 1 is a schematic of this process and following it is a short description of the main role and characteristics of each component.

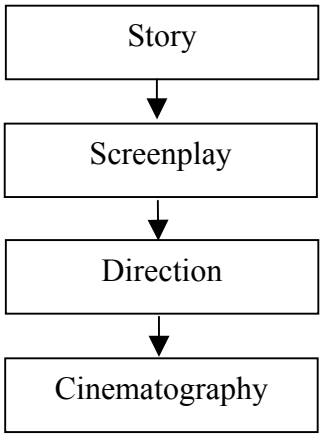

Figure 2 Film Production process

Story: Each movie is based on a Story or plot. This first step in the film production is to create a *story*. For example, the stories of many films are based on novels or comic strips. The story is a written account of the characters, the world, actions that the characters execute in the world and the effects of these actions on the world. The *Discourse* is the movie that is created from the plot of the novel. It is interesting to note here that a novel in itself is a Discourse written in natural language.

Screenplay: A screenplay is the adaptation of the story/novel/plot to suit the representation that is more presentable as a movie. There could be a long description of a rose in a novel while that is replaced by just a close up of the rose. A screenplay is a written description of how the narrative is played out on the stage.

Direction: It is a director's job to make sure the movie adheres to the specifications of the script/screenplay. A large part of the director's job is composing the scnes and getting the emotion out of the scene; considering the location, lighting etc. in context of the script.

Cinematography: Just as a screenplay writer and a director decide what to shoot, it is the job of the cinematographer to decide "how to shoot it". Study of cinematography involves the following:

Cinematic Rules: Cinematic rules have developed from practical experiences as the film industry has progressed from mute to special effects intensive, and some of them have been documented along the way. These are basic rules for 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) or Line of Interest (LOI) and relative directions of different tracking shot directions. Arijon [3] provides a very comprehensive list of all the cinematic rules that are consistently used by cinematographers.

Composition: Cinematographers tend to follow certain cinematic codes while shooting scenes. The general term for this set of codes is termed as Composition. Mise en Scene, Montage, Sound and Lighting are some codes that are important.

Continuity: This deals with choosing the shots in such a way that the viewer never feels a change in shot and there are no abrupt camera movements. Continuity should also be maintained temporally and spatially, both in the story world and the way it is presented to the viewer. Mascelli [21] gives a detailed description of continuity.

In cinematography, as mentioned above, there are certain rules that have emerged through years of practice. Certain sequences that are filmed in stereotypical ways are called cinematic *Idioms*. For example, the casual *conversation* idiom (dialogue between two players – Arijon [3] pg-51) can be viewed in three possible ways when the two actors are facing each other as shown in figures 3a and 3b.

(i) **(ii)**

Figure 3a Conversation Idiom: External Reverse Master shots (i and ii)

(i-a) External Reverse Master Shot **(i-b) Internal Shot**

(ii-a) Internal shot **(ii-b) External Reverse Master Shot**

Figure 3b Conversation Idiom: Two combinations (i and ii) of external reverse and internal camera angles

2.1.3 Camera Control in Computer Graphics

Early approaches to camera control focused on *where* to place the camera or *how* to place the camera in a virtual world. Drucker [11] defined and implemented primitive cinematic camera shots with respect to the underlying geometry in the CamDroid [52] system. This system also demonstrated the application of autonomous camera placement in different types of virtual worlds (virtual museum, virtual football game, mission planning). The UCAM system developed by Bares and Lester [52] uses a geometric constraint solver to place the camera based on the constraints set up by the user. This work has evolved into a static storyboarding tool where a user can specify the constraints on a particular frame while the system gives recommendations for camera placement.

The Virtual Cinematographer [52] introduced the concept of scripting film idioms in the form of a Declarative Camera Control Language(DCCL). DCCL was used for dynamically choosing from a set of candidate idioms specified in a film tree. The major contribution of this work was to formalize the concept of film idioms and applying it to 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. The heuristic evaluator ranks the candidate idioms on the basis of smooth transitions, crossing the line of interest, long fragments, and backward panning of the camera. Thus, the system does not capture the mood or emotion in the scene while choosing an idiom, unlike the actual director and cinematographer creating a film. Tomlinson, Blumeberg et al. [29] have recently used expressive characters driving 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.

2.2 DISCOURSE

As mentioned in the previous section, most research in computational cinematography has focused on the denotative meaning of shots. 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 consecutive shots. I draw an analogy from shots and sequences to textual sentences and multi-sentence discourse in natural language. The theory of discourse structure due to Grosz and Sidner [15] 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, the Rhetorical Structure Theory (RST) [23] 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.

2.2.1 Theory of Discourse Structure

Grosz and Sidner's theory of discourse structure [15] 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.

The Rhetorical Structure Theory, as mentioned in 2.2, captures the intentional and the informational part of discourse. The RST schemas are defined in terms of relations and they specify how text can occur coherently and relate to other text segments. RST identifies five schema types: Circumstance, Contrast, Joint, Motivation/Enablement, and Sequence/Sequence. The relations as defined by Mann and Thompson [23] are as shown in Table 1.

Table 1 RST Relation Definitions

Circumstance	Antithesis and Concession
Solutionhood	Condition and Otherwise
Elaboration	Interpretation and Evaluation
Background	Restatement and Summary
Enablement and Motivation	Sequence
Evidence and Justify	Contrast
Relations of cause	
Volitional Cause	
Non-Volitional Cause	
Volitional Result	
Non-Volitional Result	
Purpose	

2.2.2 Discourse Planning

Discourse generation systems use the theory of discourse structure discussed in the previous section to plan natural language text/utterances.

Maybury [22], in his TEXPLAN planner, formalizes the communicative acts as plan operators and implements them using a hierarchical planner. TEXPLAN uses rhetorical predicates to give a semantic classification of utterances in natural language. The surface speech acts are also classified such that they guide the selection of appropriate sentence structure using the underlying rhetoric propositional content.

The text planner for generating advisory dialogues by Moore and Paris [26] generates plans that capture both the intentional goals of the speaker and the rhetorical means to achieve them. This planner is based on the Rhetorical Structure theory discussed in the previous section. Selection heuristic for capturing the rhetorical means to achieve the intentional goals is necessary due to the fact that there are many different rhetorical strategies for achieving a given intentional 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 [19] generates stylistically appropriate text from a single representation under various settings that model the pragmatic circumstances. Pauline uses an interleaved planning and execution approach and satisfies 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 and characters are being played out in the narrative.

Longbow [54] is a discourse planner that uses a Decompositional Partial Order Causal Link planning algorithm (DPOCL) [54]. Longbow is a discourse planning algorithm that extends a partial ordered causal link planner to capture the intentional structure of discourse.

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.

The plan representation of the story is then used by a text realization component like FUF and SURGE [12,13]. FUF is a natural language generator that uses the theory of unification grammars [13]. It takes as an input a *functional description* in the form of the meaning of text to be generated, and a grammar specification. It then comes up with sentences in English that satisfy the grammatical constraints as defined by the grammar specification. This process is done in two stages: the *unification* stage, and the *linearization* stage. The input functional description is enriched with directives coming from the grammar to indicate word order, syntactic constructions etc. This enriched functional description is then given to a morphology module to handle word formation. SURGE is a surface realizing component used as a front-end to FUF and is basically a grammar of English with a large syntactic coverage.

A formal discussion of the Planning Algorithms and specifically the Longbow is pertinent as this particular algorithm is used for implementation described in the subsequent chapters.

2.2.3 Formal introduction to planning

Planning is a technique for solving problems that can be represented as having an Initial State, a Goal State and a set of operators that describe valid actions in the world. The planning algorithm represents the process of searching for a state of the world that satisfies the goal by applying a number of operators from the initial state. This process can be represented as a graph-search of all the possible connections between nodes representing the world-state after each operation. The initial and goal state are collections of predicates stating the information about the world. A plan operator is represented by:

- A precondition list
- An effect list
- A set of constraints on the operator

Formally,

Definition 1: (**Plan)** A plan P is a tuple <S, C> 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.

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 $<P_0, \Lambda, \Delta>$ where P_0 is a plan specifying the initial and the goal states, Λ is the planning problem's set of action operator definitions and Δ is the set of decomposition operator definitions.

The planning problem is solved through refinement search[27] using a ranking function that bounds the number of nodes of the plan graph and defines a pre-order on the plans. The system designer can choose to use a custom ranking function. For instance, a ranking function might be designed such that it chooses abstract actions over primitive actions. Ranking function gives is a means of controlling the search space using domain independent/dependent heuristics.

Definition 3: (**Ranking Function**) For any plan graph G, G=<n, a>, a plan ranking function *f* defines a pre-order on the plans in n.

In this thesis I use a modified version of the DPOCL[54] 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 rhetoric structure for communicative acts represented as plan operators.

Definition 4: (**Longbow Plan**) A Longbow plan is a tuple $<S, B, O, L_C, L_D>$ 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 L_C=<S_i, S_j, C> where S_i is the step whose effect establishes a precondition C of step S_j. This is illustrated in figure 3.

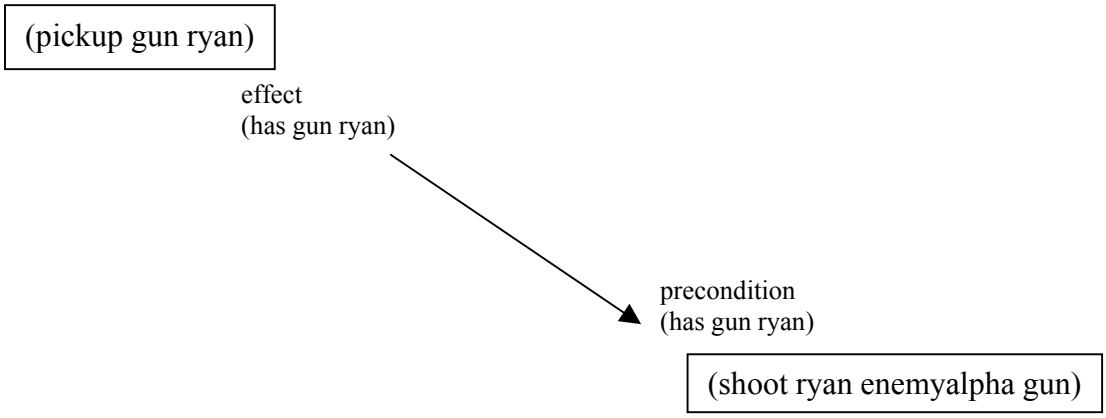

Figure 4 Causal Link

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, it is easier to encode domain knowledge as a set of abstract and primitive actions, which allows for re-use of *primitive* actions and finally it supports interleaved planning and execution.

Definition 4: (**Action Schemata**) An action schemata is a tuple <A, V, P, E, B> 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.

Primitive actions can be directly executed by the agent in the world. 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 5: (**Primitive/Composite Actions)** The set of actions $\Lambda_{prim} \subseteq \Lambda$, in a given action schemata Λ, whose members are all primitive actions. All non-primitive actions are composite actions.

Definition 6: (**Decomposition Link**) A decomposition link is a tuple $<s, s_i, s_j, s_f, s_s>$ 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.

2.2.4 DPOCL Algorithm

Longbow is a decompositional partial order causal link planning algorithm as described in Young [32] that is used for discourse planning. It uses a refinement search strategy on a plan space consisting of primitive as well as abstract actions. DPOCL algorithm models the plan reasoning of the hearer in natural language discourse and incorporates hierarchical planning directly into the causal link framework. The algorithm terminates on achievement of all communicative goals or failure to find further actions for achieving a goal. It chooses non-deterministically to either expand an abstract action or tries to add an action that satisfies an unachieved goal. The actions library consists of primitive and abstract actions with decomposition specification (Figure 5).

```
(define (action DESCRIBE)
    :parameters (?X)
    :primitive NIL
    :description NIL
    :precondition NIL
    :effect ((KNOW-ABOUT-HEARER ?X))
    :constraints NIL)

(define (decomposition DESCRIBE)
    :parameters (?OBJECT)
    :constraints ((OBJECT ?OBJECT) (CLASS ?OBJECT ?CLASS)
                  (HAS-PARTS ?OBJECT ?PARTS))
    :description NIL
    :links ((STEP1 (KNOW-ABOUT-HEARER (CLASS ?OBJECT ?CLASS)) END)
            (FORALL ?PART IN ?PARTS
```

```
              (STEP2 (KNOW-ABOUT-HEARER (HAS-PART ?OBJECT ?PART)) END)))
    :steps ((END (FINISH ?OBJECT)) (BEGIN (START ?OBJECT))
            (STEP1 (DESCRIBE-CLASS ?OBJECT ?CLASS))
            (FORALL ?PART IN ?PARTS
             (STEP2 (DESCRIBE-HAS-PART ?OBJECT ?PART)))
            (STEP3 (SUMMARIZE-CLASS ?CLASS)))
    :orderings ((STEP1 STEP2))
    :rewrites (((KNOW-ABOUT-HEARER ?OBJECT)
                ((KNOW-ABOUT-HEARER (CLASS ?OBJECT ?CLASS))
                 (FORALL ?PART IN ?PARTS
                  (KNOW-ABOUT-HEARER (HAS-PART ?OBJECT ?PART)))
                 (KNOW-ABOUT-HEARER (SUMMARY ?CLASS))))))

(define (action DESCRIBE-CLASS)
    :parameters (?X ?CLASS)
    :primitive T
    :description NIL
    :precondition NIL
    :effect ((KNOW-ABOUT-HEARER (CLASS ?X ?CLASS)))
    :constraints NIL)
```

Figure 5 Abstract and primitive action definition in Longbow

A plan produced by Longbow is a data structure of partially ordered steps with variables bound to the objects in the universe of discourse. A partial ordering is enforced through causal links and binding constraints on the steps in the plan. A detailed description of the plan data structure as generated by the story planner is given in section 3.1.2.

3. MIMESIS FILM SYSTEM

Nature allows only experimental situations to occur which can be described within the framework of the formalism of quantum mechanics

-Werner Heisenberg

The Mimesis system[30] is an architecture and a collection of tools integrating the high-level intelligent control with a range of virtual environments ranging from PDAs to 3D game engines. Mimesis is described in more detail in the following section. The mimesis FILM system is built on the mimesis architecture and has been designed to model the film hierarchy discussed in the chapter 2. At the core of the FILM system is the camera planner that uses the modified DPOCL algorithm to generate a camera plan structured to convey to the user the story generated by a story planner. Film idioms are formalized as plan operators with primitive camera actions (e.g., track, pan, look-at, over the shoulder, dolly, tilt) representing the ground operators of the planning problem. The input to the camera planner is a declarative description of the story plan, including information about the steps in the plan, the temporal and causal relationships between them, and a set of additional geometric and spatial constraints on the camera. Abstract camera operators or *episodes,* representing film idioms, are decomposed by the camera planner into candidate sequences with episodes and/or primitive actions. These candidate plans are then evaluated and ranked by the ranking function and the top ranking plan could be communicated to the underlying graphics engine, Unreal Tournament 2003[1] through a socket connection; the graphics engine then uses the specification of actions in the plan steps to control the player's camera as the story is also played out within the virtual world.

[1] Throughout this document we are using the game engine Unreal Tournament 2003 for visualization. However, the camera planning algorithm is a general algorithm that is independent of visualization programs.

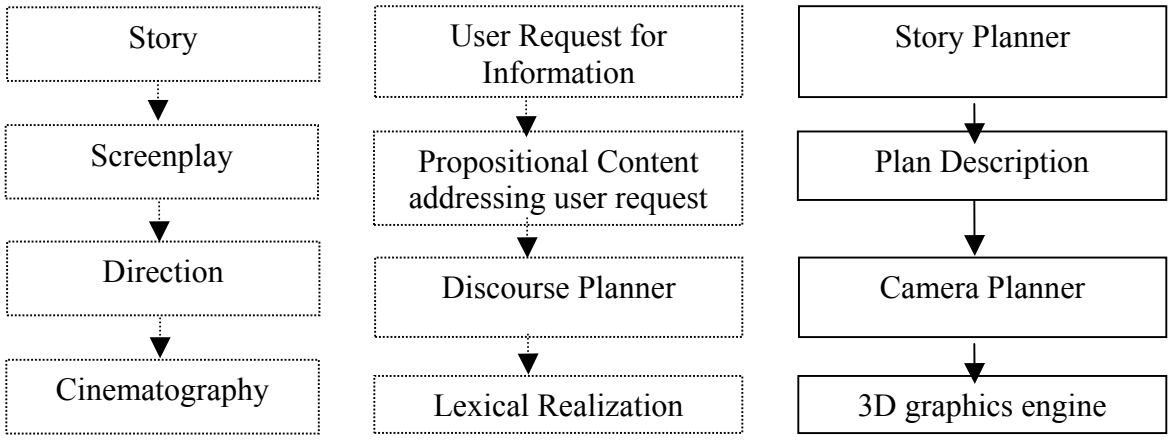

Figure 6 Parallels between Cinematic production, Discourse plan production and the Mimesis film system architecture

As shown in Figure 5, the mimesis film system parallels the film hierarchy that was described in section 2.2.1. By structuring the system this way, we can more readily integrate models of the well-established practices of filmmaking to generate cinematic discourse to convey a given story. This type of layered architecture also gives us the flexibility to choose the type of representation at each layer of implementation. For instance, the camera primitives implemented on the graphics engine could be scripted independent of the camera plan specification. In the following sections, a detailed description of the mimesis architecture is given, followed by a discussion of the representation used for the story world plan by the camera planner. Next, the formal description of the algorithm used for generating the camera plan is presented, and finally we discuss the details of the Unreal Tournament implementation.

3.1 MIMESIS Architecture

Mimesis is a system developed with the intention of integrating AI control with a wide range of virtual environments. It uses a client/server architecture in order to provide consistent representation of objects, actors, actions and events across different virtual environments with differing procedural representations of action and change. The Mimesis server components perform the high level narrative control over the virtual

environment and user interaction, while customized clients handle the low-level interaction. The main components of the mimesis architecture are illustrated in figure 6 and are described briefly in the following sub-sections.

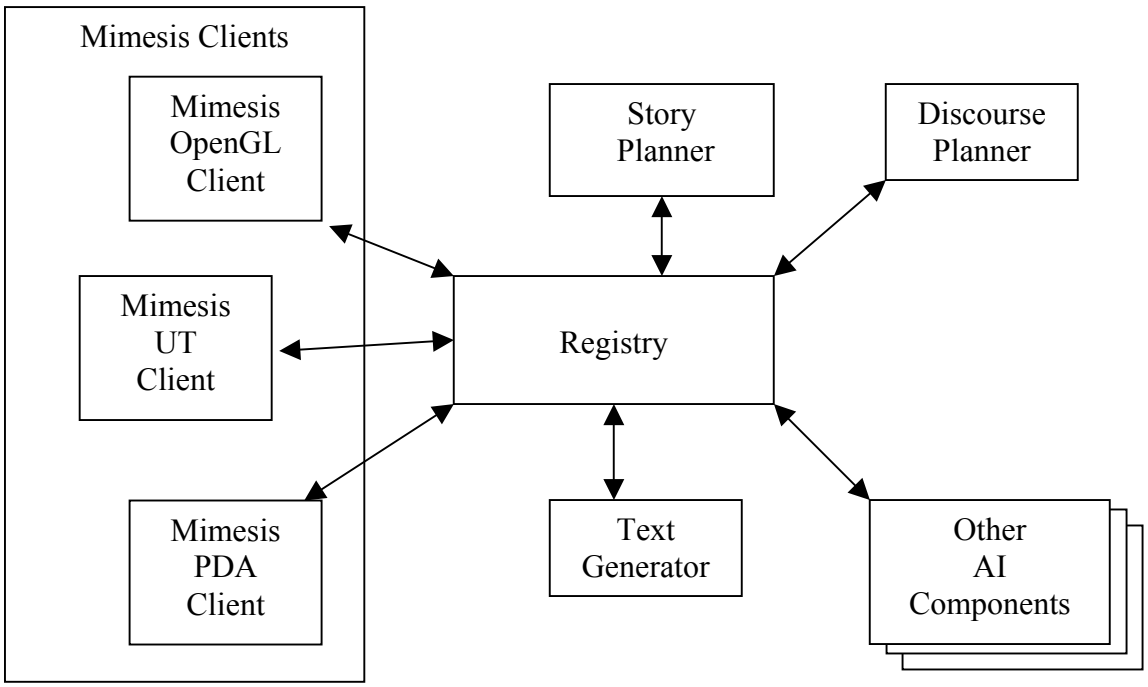

Figure 7 Mimesis Architecture

3.1.1 Mimesis Registry

The primary function of the Mimesis Registry is to handle communication between the server modules and the clients. Mimesis has a highly modularized architecture, with all the modules communicating via a well-defined XML-based messaging protocol among each other and with clients. At system startup, each component registers the message types that it can handle with the central registry that then controls the execution of the distributed components. The components of Mimesis most relevant to this thesis are the story planner and the discourse planner. The story planner module generates stories in the form of partially ordered set of actions of characters in the story world. The discourse planner comprises a text component and a camera component. The discourse planner generates communicative acts for the plans generated by Longbow. Other comonents

include a text generation system, a user model and an HTTP server. The mimesis components relevant to this thesis are briefly described in the following sections.

3.1.2 Story Planner

The story planner module is written in Lisp and uses the Longbow planning system for generation of partially ordered specification for actions in a story world. It takes as an input 1) a declarative representation of all the actions that can be performed in the virtual world 2)a description of the initial state of the story world and 3) the goals for the story. The Longbow planner uses the DPOCL algorithm (described in detail in section 2.2.3). to generate a Directed Acyclic Graph(DAG) [Figure 8 Plan DAG] representing the story's actions and their temporal orderings. The client communicates all the interaction back to the server and the story planner. Figure 8 shows an example of a story represented as a DAG. In this figure, story actions are indicated by rectanges and are labeded with the act-type and arguments of the corresponding action. The conditions above each action indicate the preconditions for the actions, and the arrows drawn from one action's effects to the preconditions of another action indicate causal links that connect the source effect with the destination preconditions. Temporal ordering is indicated in a left-to-right manner.

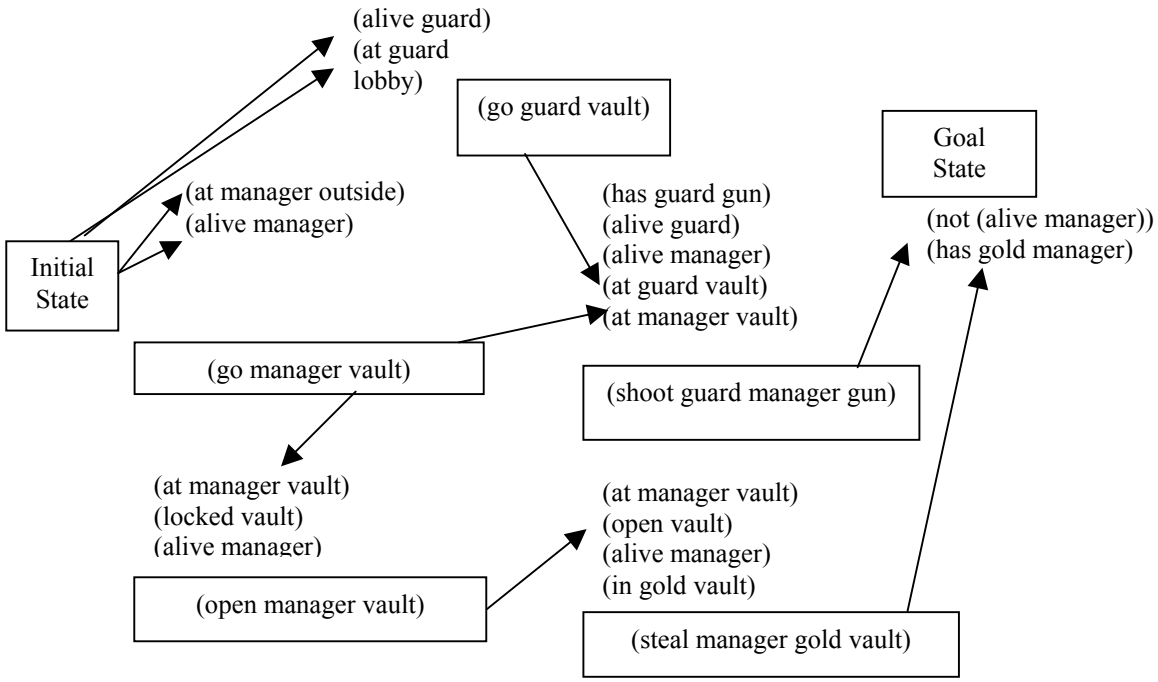

Figure 8 Plan DAG: In this figure the boxes represent storyworld actions, the predicates above each box are the preconditions of the action and the arrows denote links from the effects of an action to the consequent precondition of another action that is causally linked to it.

The plan data structure for a narrative planner is similar to the one used for discourse planning. The main components of the data structure that are relevant to this work are briefly described here:

- **Steps**: Each plan has a list of steps that represent the character actions with variable bindings from the objects in the story world. A step data structure also has a list of *preconditions*, *effects* and *constraints* on the variable *bindings*. A step is identified by its *act-type* and it has a Boolean variable indicating whether it is a *primitive* step or not. The step also has links to its parents and descendants in the plan graph.
- **Bindings**: A list of all the variable bindings for each step indexed by the step numbers and the binding constraints on the variables.
- **Links**: Stores the source, target and condition of all the causal links in the plan.

3.1.3 Generating Plan Description Predicates

The story planner generates a plan data structure, which is a partially ordered set of actions for the characters in the world. This data structure is then converted into a representation generated by an intermediate module consisting of description predicates as shown in Figure 9. The representation contains information about the act-type, preconditions, effects, orderings, constraints and causal links with all the bindings associated with each step. We follow the plan representation language defined by Fergusen[14]. The camera planner uses not only information about objects, actions and events in the story world but also temporal relationships between actions and events taking place. This information is used in determining camera transitions and shot lengths (refer chapter 5 for an example). In order to temporally index the facts about the story at different points in time from the story world plan, we use the temporal intervals defined by Allen[1] as shown in [Table 2 Allen's temporal intervals].

(HOLDS (Before ?step0) ?step1))	?step0 executes before the start of ?step1
(HOLDS (Equals ?step) ?condition))	?condition is true only during the execution of ?step
(HOLDS (Meets ?step0) ?step1))	?step0 executes immediately before the start of ?step1
(HOLDS (During ?step) ?condition))	?condition negates during the execution of ?step
(HOLDS (Overlaps ?step0) ?step1))	the execution of ?step0 and ?step1 overlaps
(HOLDS (Start ?step) ?condition))	?condition negates at the start of ?step
(HOLDS (Finish ?step) ?condition))	?condition negates at the end of ?step

Table 2 Allen's temporal intervals (the other 6 relations out of 13 defined by Allen are inverses of Before, Meets, During, Start, Finish and Equals)

```
(ACT-TYPE STEP6 OPEN-VAULT)

(PRECONDS STEP6 (AND (ALIVE PREZ) (AT PREZ VAULT1) (LOCKED VAULT1)
(KEY-OPENS KEY1 VAULT1) (OPENABLE VAULT1)))
(HOLDS (MEETS STEP6) (ALIVE PREZ))

(EFFECTS STEP6 (OPEN VAULT1))
(HOLDS (AFTER STEP6) (OPEN VAULT1))

(CONSTRAINTS STEP6 ((CHARACTER PREZ) (VAULT VAULT1) (KEY KEY1)))
(HOLDS (EQUALS STEP6 (CHARACTER PREZ)))

(NECESSARILY-BEFORE STEP6 STEP5)
(HOLDS (BEFORE STEP6) STEP5)

(LINK STEP6 STEP0 (ALIVE PREZ))
(HOLDS (BETWEEN STEP0 STEP6) (ALIVE PREZ))
```

Figure 10 Story world action description

This knowledge about the facts of the story world indexed temporally with respect to other actions and events in the story is used by the camera planner to determine the temporal ordering of the camera shots. The camera planner queries this knowledge base through an interface for specific temporal orderings and constraints on the story world plan. This information is required for decomposition of episodes and selection of appropriate idioms. For example, the episode decomposition for a conversation searches an (act-type? Step Speak) that returns all the steps with speech acts for filming the conversation. This representation is also useful for determining what facts in the story world hold true during the execution of a particular story world action. The temporal ordering of steps in the story world also determines the selection of camera primitives and this is used by the planning algorithm for determining shot boundaries. The selection of episodes for decomposition and camera primitives is described in detail in the chapter 4.

3.1.4 Camera Planner

The discourse planner is the main focus of this thesis. There are several modules for different modes of discourse generation in the mimesis system. The camera planner generates visual discourse. It uses the Longbow planner with a modified DPOCL algorithm to generate sequences of camera shots or primitive communicative actions for

conveying the story generated by the story planner to the viewer. The discourse planner takes as an input a story world plan and a set of temporal and spatial constraints along with cinematic communicative goals. The output is a directive for 3D actions for execution within the virtual world.

The camera planner takes in as an input a story world plan data structure described in the previous section and information about the state of the world using the plan representation defined by Ferguson [14]. The planning algorithm generates a sequence of primitive camera steps to satisfy the communicative goals. The communicative goals like their natural language counterparts, are of the form (Knows ?viewer ?time P). The camera plan is also communicated through a socket to the mimesis client where it is executed in the respective virtual world implemented by the client.

3.1.5 Mimesis Clients

Mimesis clients are different types of virtual worlds that serve as test beds for integration with the AI server components. Since all the AI control uses planning techniques, the mimesis clients are designed to execute plans communicated to them through a socket connection. The main function of an execution manager component in the clients is mapping the action descriptions to executable functions implemented in the virtual world (e.g. character actions or communicative actions). During the mapping process the client associates world objects with object names specified in the action description.

As mentioned earlier, the test bed for this particular work is the Unreal Tournament 2003(UT) engine. The purpose of choosing this engine was two-fold. Firstly, the story planner already worked well with the UT engine and the action class architecture was also already built into the engine. Secondly, due to the fact that it is a game engine, it would be easier to test the camera control in interactive applications in future. The UT engine has a mimesis *execution monitor* that 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 *MfilmPlayer* in the *MFilmGame* 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 player's camera. The player's view is updated after getting recommendation for the camera position from the cinematographer, who takes into consideration the currently set up constraint values for updating the camera location. The architecture of the visualization module in Unreal Tournament 2003 game engine is shown in Figure 11.

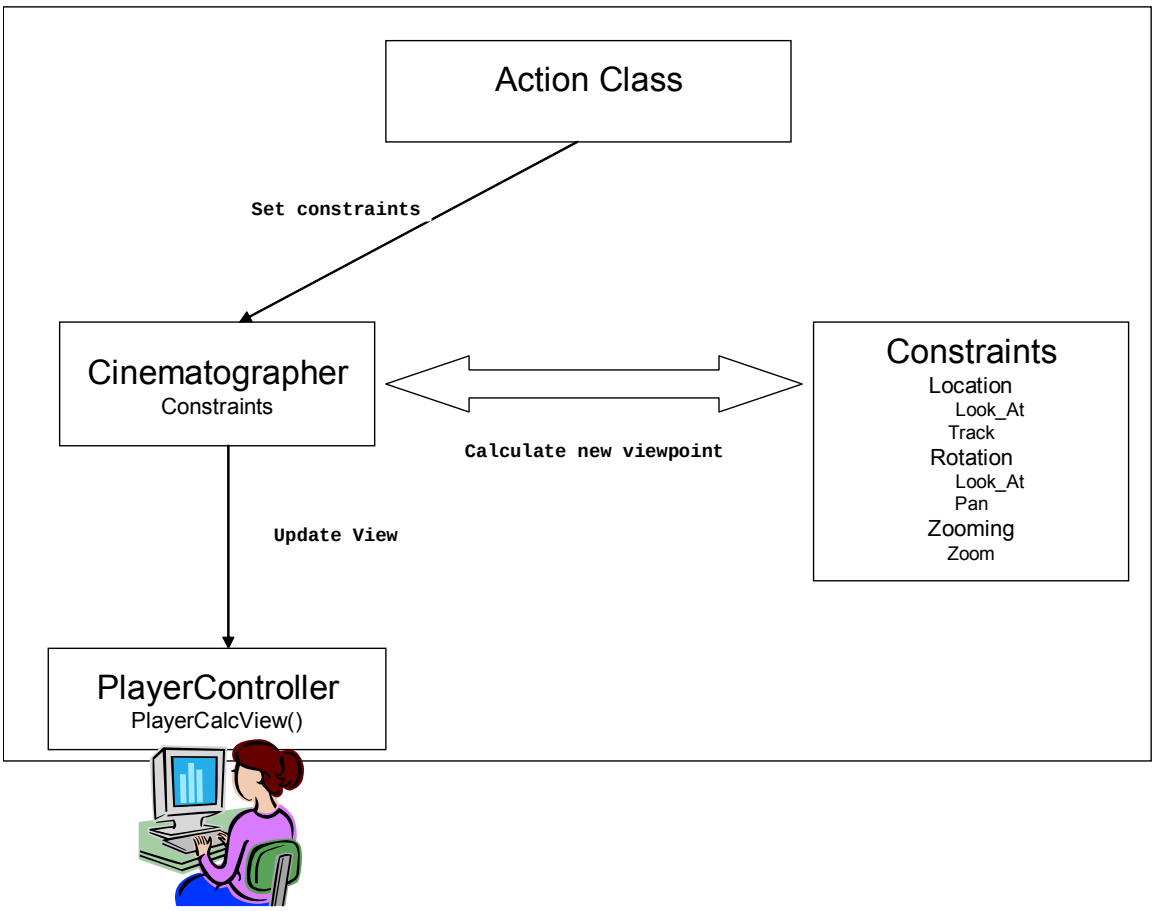

Figure 11 Architecture of the visualization module in Unreal Tournament 2003

4. A Discourse Planning Approach to Camera Control

It is the **theory** which decides what we can **observe**.
-Albert Einstein

4.1 Formal Preliminaries

Continuing on the formal description of the discourse planning problem, we state the problem of shot and sequence composition as the problem of *generating a list of partially ordered steps specifying abstract and primitive camera actions that change beliefs and mental states of the viewer in order to achieve a set of communicative goals*. The input to the planning problem is the story world plan with the facts about the plan's actions temporally indexed by step numbers in a declarative representation described in Section 3.2 that follows [14]. Additional spatial and temporal constraints specific to the geometry representation are added for use by the camera operators. The primitive actions generated by the camera planner are then mapped to action classes and executed on the Mimesis game client.

Formally,

Definition 7: (**Camera Planning Problem)** The camera planning problem is represented by the tuple $<P_S \cup W, A, I, U, B, G>$ where P_S is the set of sentences describing the plan data-structure for the story world plan, W is the set of sentences describing a) facts about the story world temporally indexed as description predicates as described in the previous chapter and b) additional facts about the story world not included in the story plan. These additional facts may include annotations characterizing features of the story world plan (e.g., its tempo at various points during its execution) or desired features of the camera plan (e.g., the mood of the camera plan at particular points during the story world plan's execution). A is the set of primitive and abstract action operators for the camera, I is the set of predicates indicating the initial knowledge of the viewer and G is the set of communicative goals for the camera planner. U is the finite universe of discourse for the variables over the objects, actions and events within the story world and B is the set of

binding constraints over variables in the camera operators. Initially, B is set to the empty set.

Definition 8: (**Camera Action)** A camera step/action is defined as <P, E, A> where P is the set of preconditions on the operator, E is the set of effects and A is the set of assertions on the story-world steps.

Abstract camera actions are called *Episodes*. Episodes can be decomposed into different idioms for filming the same act-type. For example, a Conversation episode could be decomposed using either of the two idioms shown in Figures 3A and 3B.

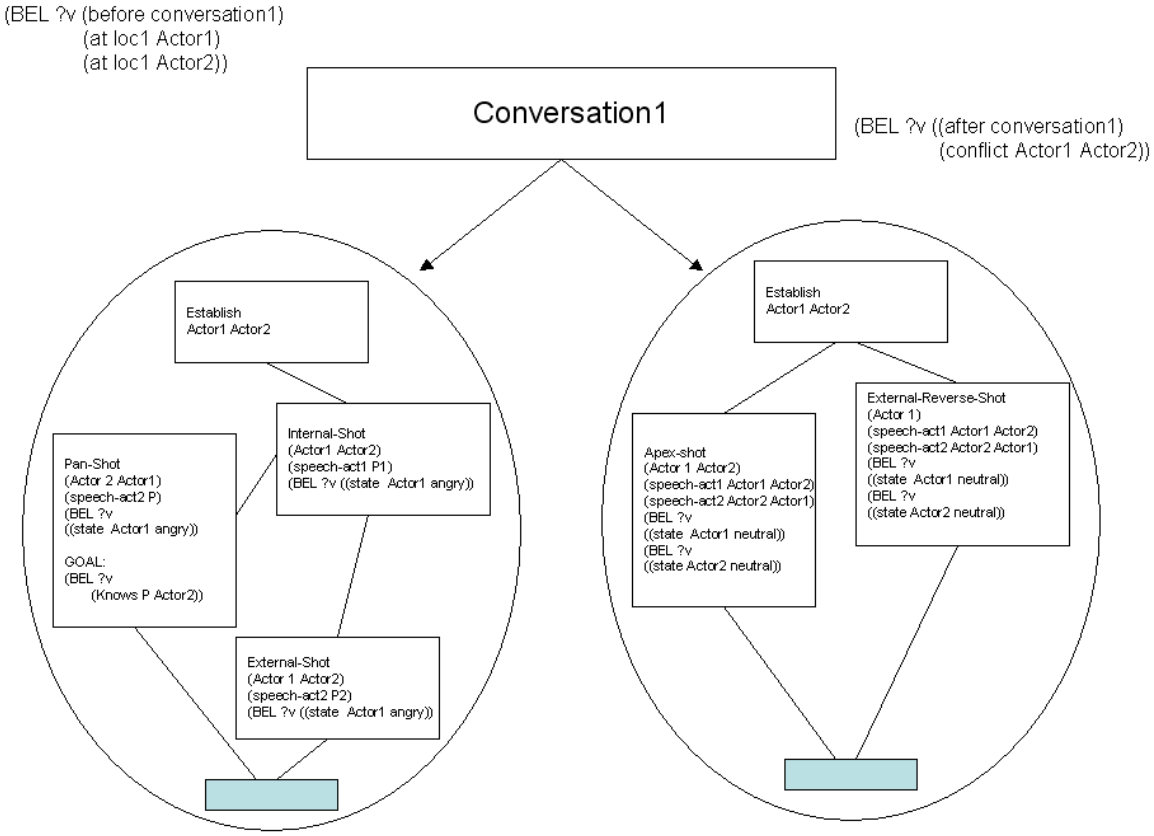

Figure 12 Illustration of decomposition of a conversation idiom (The notation in the figure is simplified for better readability). Here, dashed arrows indicate alternate decompositions. Solid arrows indicate temporal ordering among sibling steps.

A critical aspect of camera planning is the relative temporal sequencing of camera and story world actions. This is accomplished by links from the camera plan steps to the beginning and end of the story world steps that the camera step films.

Definition 9: (**Assertive Link)** An assertive link is defined as a tuple <ϕ, ψ, E> where ϕ is of the form (Begin (at-start S_i)), (Begin (during S_i)) or (Begin (at-end S_i)) and ψ is of the form (End (at-start S_j)), (End (during S_j)) or (End (at-end S_j)), here S_i and S_j are storyworld actions.

The assertive links describe the relative ordering of story world actions and camera actions. Here, (at-end S) indicates the time point at which story world step S terminates execution, and (at-start S) indicates the time point at which story world step S initiates execution and (during S) indicates any time between the initiation and termination of S (including its start and end points). (Begin x), where x is one of the at-start, at-end or during forms described above, indicates that the camera action must start at the point or during the interval determined by x. (End x) indicates that the camera action must terminate at the point or during the interval determined by x.

4.2 Viewer Model

In our approach, the execution of a plan created by an autonomous camera planning system is modeled as an interaction between the camera and a viewer. The planning algorithm is implemented for the camera agent to affect the viewer's beliefs about the story world with cinematic goals representing the communicative intentions of the director. The plan operators encode the cinematic knowledge that is used for conveying cinematic meaning using the language of film idioms; these operators are selected by the camera planner based on their relation to the underlying story world actions that they film. A similar approach is taken for modeling agents in natural language processing/generation systems for interacting agents where agents use speech acts to change the beliefs of other agents. Modeling the visual primitives as communicative acts raises the same issues as the ones raised by natural language researchers for generation of speech acts for dialogue agents. Some of the issues that need to be studied when classifying visual primitives are outlined below,

1. **What is the intended purpose of the communicative act**: The purpose of a cinematic communicative act for a camera planner is to convey the states of the storyworld to the viewer at a certain time during the progression of the story. The state of the storyworld contains information about the actors, objects, events and execution of actions occuring within the story world.
2. **How can the propositional content of the communicative act be conveyed**: The propositional content that is conveyed to the viewer by either, characters or objects in the story or by extradiegetic narrator, in addtion to the visual setup of the world that is framed by the camera.
3. **What is the current focus of attention in the ongoing discourse:** In generation of discourse each discourse segment directs the hearer/viewer's attention towards the focus of the information contained in the segment. The choice of cinematic idiom is also influenced by the focus of attention for the viewer within the story world.
4. **What point of view is to be used:** In communicating a story there might be multiple actors or objects within the world. The choice of filming the story world from a 3rd person camera or from the point of view of characters or objects within the storyworld has a significant effect on the viewer.
5. **Rhetorical relation between acts:** The primitive communicative act also conveys the communicative goals of the director by influencing the viewer's model based on the context established by previous communicative acts in the discourse.

In this thesis, we address questions 1, 2 and 5. The issues identified in Questions 3 and 4 are beyond the scope of this work. We assume for purposes of the thesis that information about point of view and mood of the story world plans is included in W, a portion of input to the camera planning problem, and is thus accessible to the plan construction process. This information is currently assumed to be provided by some system or human exogenous to the planning system.

In Figure 14, a classification of communicative acts is shown based on the film idioms that the operators use to communicate the story. The mood of the scene, number of characters, spatial set up of the scene and rhetorical relationships of actions to other actions in the discourse are the parameters used to classify the communicative actions that drive the camera. A parallel for such a classification scheme for speech acts in explanation generation systesms can be found in Maybury[53]. This classification is consistent with the hierarchy of communicative acts for discourse generation where the top level rhetorical actions along with the communicative intentions of the director and the state of the world drive the selection of primitive actions.

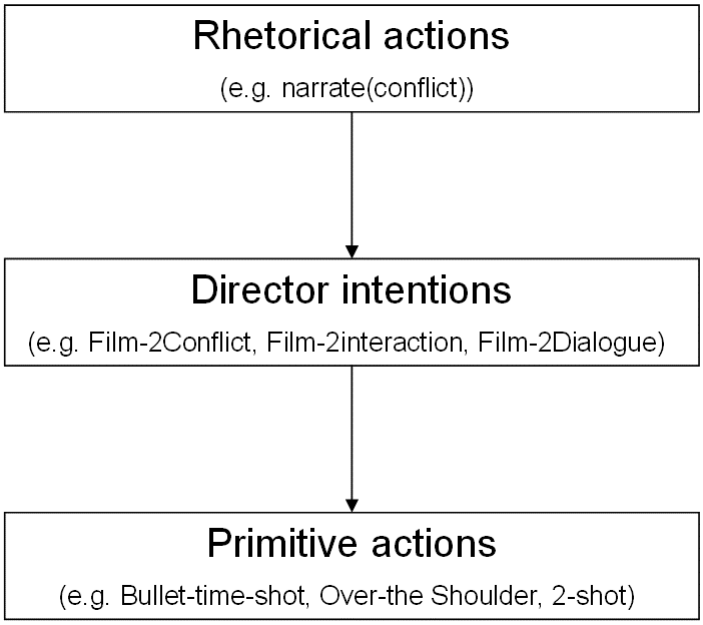

Figure 13 Hierarchy of communicative acts and their role at each level of generation.

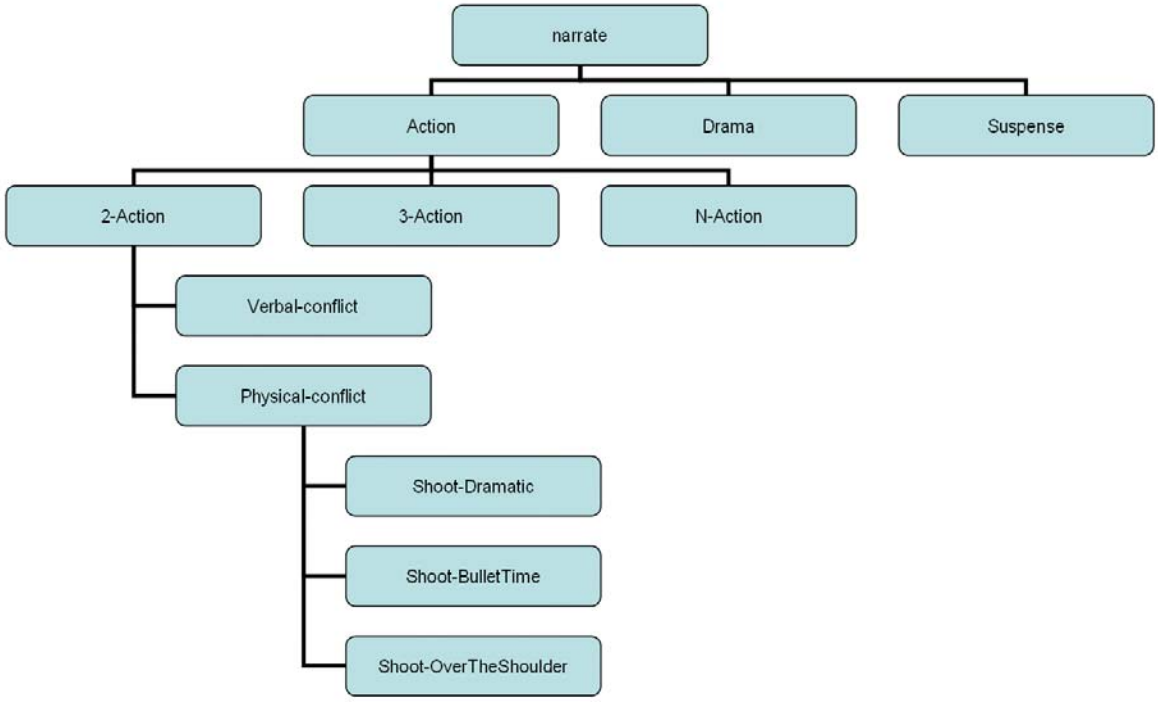

Figure 14 Example classification of visual communicative acts. The classification of operators is based on the film idioms that are commonly used for the corresponding type of action occurring in the story world.

The model of the viewer used in this work represents the viewer's knowledge about the story world actions, events and objects temporally indexed relative to other actions, events or objects. This representation of the viewer model captures both the rhetorical structure of the narrative that is being conveyed at a higher level, and the localized effects of the primitive actions and story world actions that affect the viewer. This representation also aids the camera planner in reasoning about the manipulation of the viewer model and selection of shots to take advantage of expressing emotions, suspense, etc. This representation thus allows for a very expressive vocabulary on the camera planner's part. The assumption that a particular shot selection will successfully achieve its intended effect is enforced here since it is beyond the scope of this thesis to study and evaluate the effects of similar primitives on different viewer models. This remains a common pragmatic issue even in natural language generation systems.

The beliefs of the viewer are represented in the form:

(BEL ?V ?T ϕ)

Where, ϕ is the statement about the storyworld that the viewer ?V believes to be true at time ?T in the story.

The viewer's knowledge base contains beliefs about 5 types of storyworld information:

- **Spatial Knowledge**

  ∀T (BEL ?V ?T (type ?object ?type))

  ∀T (BEL ?V ?T (type act-1 action))

  ∀T (BEL ?V ?T (type loc-1 location))

  (BEL ?V ?T (occurs t1 act-1))

  (BEL ?V ?T (at loc-1 act-1))

  Viewer's spatial knowledge about locations, objects and actions in the story world.

- **Participant Knowledge**

  ∀T (BEL ?V ?T (type actor-1 actor))

  ∀T (BEL ?V ?T (type actor-2 actor))

  (BEL ?V ?T (lover actor-1 actor-2))

  The knowledge about actors, events, objects and their relationships with other actors, events, objects in the story world.

- **State Knowledge**

  (BEL ?V ?T (state actor-1 ecstatic))

  (BEL ?V ?T (state actor-2 sad))

  The knowledge about the state of the world and the emotional state of the participants.

- **Temporal Knowledge**

  (BEL ?V (during act-1) (state actor-1 tense))

  (BEL ?V (finish act-1) (state actor-1 ecstatic))

  The temporal information about the story world that the viewer has is expressed as relative predicates defined by Allen[1]. In this case *time-pv* belongs to the set {before, meets, during, …}

- **Viewer's state and overall mood and tempo of the narrative**

  (BEL ?V ?T (type ?act-2 football-game))

  (BEL ?V (during act-2) (tempo fast))

  (BEL ?V (finish act-2) (tempo slow))

  (BEL ?V (during act-2) (mood tense))

  (BEL ?V (finish act-2) (mood happy))

4.3 Algorithm

The camera planner uses a modified version of the DPOCL refinement search algorithm described in section 2.2.4. Figure 11 shows an outline of the algorithm.

The camera planning algorithm takes as an input three elements: a description of the storyworld plan, a set of goals for the camera planner and a description of the starting state of the camera plan. The story world plan is encoded in the representation described in Chapter 3. The goals of the camera planner are represented as a description of the mental state viewer – in particular, the viewer's beliefs about the story world and the story world plan. The camera planner, given the story world actions, events and objects, constructs a plan of camera actions whose cumulative effects modify the viewer's beliefs to reflect the planner's goals. The model and the process used to create the camera plan is analogous to the approach used by natural language discourse generation systems when constructing plans to achieve discourse goals.

Cam-Plan (P_C = <S, B, O, L_C, L_D, L_A>, Λ, Δ)

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 a single ordering constraint between them requiring the initial state step to preceed the goal state step.

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 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 camera step S' asserts e then update the assertive link such that T(S')+<T(S_{wi})<T(S')-. If no existing camera step asserts e then add a new step S_{add} and update the causal and assertive links <S_{add}, e, p, S>. S=S∪S_{add}, L_a=L_a∪<S_{wi}, S_{wj}, T, S_{add}, e)
2. **Episode Decomposition**
    a. **Action Selection**: Non-deterministically select an unexpanded episode from P_C
    b. **Decomposition Selection**: Select an idiom for the chosen episode and add to P_C the steps and constraints specified by the operator as the subplan for the chosen episode.

Conflict Resolution

Two assertive links A_1 = <ϕ_1, ψ_1, E_1> of step S_1 and A_2 = <ϕ_2, ψ_2, E_2> of step S_2 *conflict* just when A_1 and/or A_2 indicate that the execution of S_1 and S_2 might possibly overlap. In this case since it is not possible to have two camera actions filming a common storyworld interval, additional ordering constraints are enforced on the storyworld actions for determining the ordering of camera actions.

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

Promotion: Move S_1 before S_2

Demotion: if S_2 before S_1

Temporal Separation: add temporal ordering constraints on the two actions C_i and C_j

A step S_t *threatens* the causal link L =<S_i, S_j, C> if S_t might possibly occur between S_i and S_j and S_t asserts a condition as an effect that unifies with C. For each threat in P_C created by the causal or episodic planning above, resolve the threat by nondeterministically chosing one of the following procedures:

Promotion: Move S_j before S_t if the ordering constraints are not violated

Demotion: if S_t before S_j if the ordering constraints are not violated

Variable Separation: add variable binding constraints to prevent the relevant conditions from unifying

Recursive invocation

Call Cam-Plan with the new value of P_C.

Figure 15 Sketch of Camera Planning Algorithm

4.3 Integrating the execution of camera and story world actions

In this section, I describe how the story plan is linked to camera actions through assertion links based on the classification of camera actions and their relationships with story world actions.

The camera planner generates plans for camera steps that are connected to the story world actions through links that establish the relationship between the camera actions to the corresponding story world actions. It generates all possible combination of shots for viewing a story. The ranking function is then used to rank the best combination based on heuristics described in the following section. The links between story world actions and the camera actions are used to determine temporal threats in the ordering of story world actions or camera actions and are used for combining sequences of camera actions to generate smooth transitions. In the cases where the beginning and ending links overlap, additional ordering constraints are enforced on the storyworld actions and the camera actions to resolve the conflicts. The abstract camera actions are decomposed into primitive actions that film a group of story world actions. For instance, the conversation idiom is used to film a group of speech acts in the story world. There are certain parameters that determine how the story world is segmented into groups of primitive actions that are filmed by abstract actions for the camera planner.

The parameters used in deciding the continuity of segments of story world actions are:

- Number of participants
- Spatial continuity
- Temporal continuity
- Rhetorical relationships between story world actions

Once the camera action is selected based on the above mentioned factors, the abstract actions are decomposed into candidate sequences.

4.4 Heuristics for selection of primitives

The DPOCL planner on which we build our camera planning algorithm uses an approach to plan generation called plan-space search. In plan-space search, the planning process procedes by searching through a directed acyclic graph representing a space of possibly partial plans. Nodes in the graph represent possibly partial plans. Arcs between a parent and a set of children nodes indicate that the parent node was incrementally refined to create the plans corresponding to each of the children nodes. The root node of the graph represents the empty plan, and leaf nodes represent plans that are either complete (i.e., those that are flaw-free), inconsistent (e.g., those that have inconsistent temporal constraints) or remain indeterminiate (i.e., have not yet been expanded during the search).

To guide the search process during planning, the DPOCL algorithm uses best-first approach. A heuristic search function ranks indeterminant nodes at the fringe of the search space, ordering the unexpanded nodes most-promising to least-promising. In our approach, we take advantage of the heuristic search function to rank plans not only based on estimations of how close the plans are to being complete, but also based on the match between the structure of the plan and desirable features of the plan's narrative structure. Specifically, he selection of primitives for a decomposition takes into consideration the parameters summarized in Table 3 that affect the selection and parametrization of primitives shots.

Heuristic	**Effect**
Tempo	The tempo of the scene determines the shot length and the number of shots required to film a particular sequence.
Motion	Shots can be classified into one of 5 categories based on how the motion of the camera and actors occurs as perceived by the viewer • Single Location, Static Characters • Single Location, Internal Motion(within the frame)

	• Multiple Locations, Static Characters • Multiple Locations, Internal Motion(within the locations) • Single/Multiple Locations, External Motion
Mood	The mood parameter applies to • mood of the scene • mood of the characters involved • mood of the viewer at that time (from mood of the previous scene)
Spatial Expanse	The spatial expanse of the location and relationship with other locations.
Temporal Expanse	Variations in the time as perceived by the viewer.

Table 3 Heuristics that affect shot selection

Details of the way that these parameters affect the selection of primitives follow below.

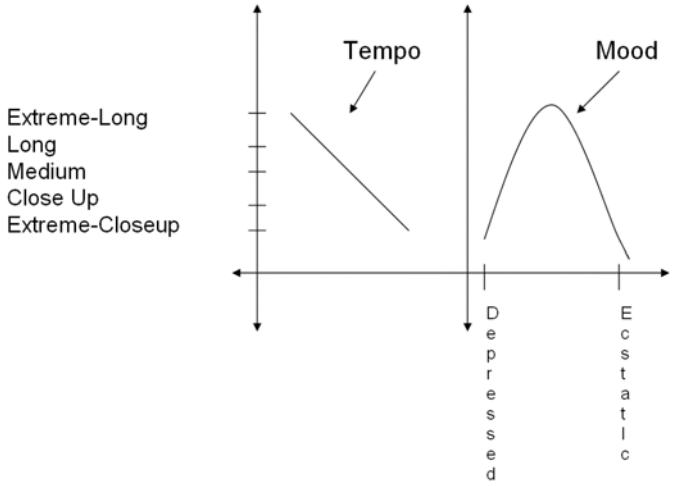

Figure 16 Relationship between selection heuristics and primitive shots

Tempo

Communicating a story involves describing/narrating objects, actions, events etc. in the story world and their changes over time. The percieved rate of flow of time in a narrative, or the narrative's *tempo*, is controlled by the storyteller in a variety of ways. Cinematographers have used and developed standard techniques for controlling tempo. For instance, tempo can be slowed down by choosing static long shots that film less motion within the frame, and can be increased by using rapid cuts with close-up shots and reduced shot lengths.

Computer vision researchers[6] have attempted to annotate video frames to determine the relationship of tempo with shot length and movement of the camera. The following equation is derived from the initial attempts at video annotation.

$$P(n) = \frac{\alpha\,(med_s - s(n))}{\sigma_s} + \frac{\beta\,(m(n) - \mu_m)}{\sigma_s}$$

Where,
s = shot length
m = motion magnitude
μ = mean

Figure 16 Relationship between Tempo, Shot length and motion of the camera

Mood

The mood of the scene and the mood of the characters involved in the scene also determines the placement of shots and type of shots used for conveying the emotions. The relationship between Mood and shot-types is shown in Figure 15.

Motion

As shown in Table 3 Motion is classified into 5 categories to determine the shot type based on the spatial and temporal relationship between the framed actors and external actors.

Spatial Expanse

The Spatial expanse alongwith the geometric parameters of the characters involved affect the panoramic shots and establishing shots. They could also be used for determining shot lengths and potentially for interleaving camera actions during the execution of long spatial movement of characters, especially in real-time game environments where characters have to actually span distances.

Temporal Expanse

Time in a story can range from seconds to years, and there are specific timing constraints on a movie or a video game for conveying the complete story. The selection of shots, especially transitions are used as cues for conveying temporal changes in the story world.

In the current system we use a very primitive heuristic function for determination of shots.

5. Example Story

An ounce of practice is worth a pound of preaching.
-Proverb

The example story that we use to illustrate the camera planner's functionality centers on the protagonist, Ryan, a Navy pilot whose plane has crashed behind enemy lines. Ryan's goal is first to recover from his crash site a computer disk containing reconnaisance imagery and then to escape into the nearby mountains. One linearization of the story plan that is generated by the story planner for this story is shown in Figure 17.

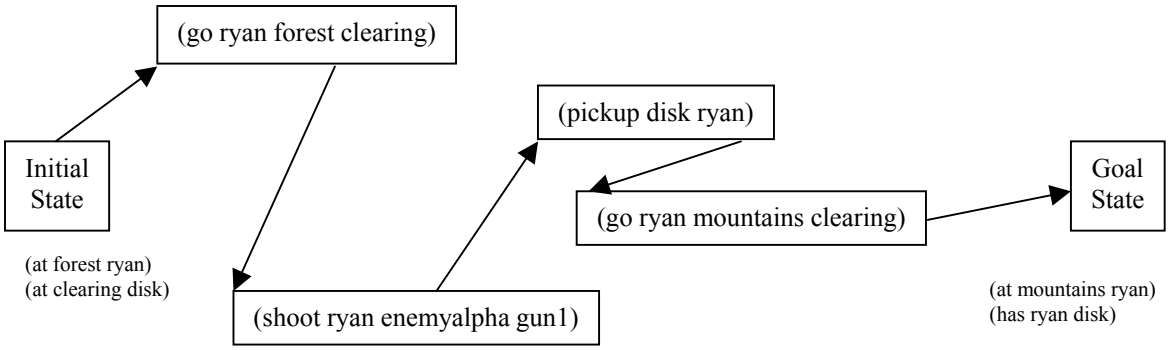

Figure 17 Linearization of a story plan.

This story plan is input to the module that generates a declarative description of the actions in the story plan. A simplified readable version of the resulting description is shown in Figure 18.

(ACT-TYPE STEP3 SHOOT)

(PRECONDS STEP3 (AND (ALIVE ENEMYALPHA) (AT RYAN CLEARING)))
(HOLDS (MEETS STEP3) (ALIVE PREZ))

(EFFECTS STEP3 (NOT (ALIVE ENEMYALPHA)))
(HOLDS (AFTER STEP3) (OPEN VAULT1))

(CONSTRAINTS STEP3 (CHARACTER RYAN))

(NECESSARILY-BEFORE STEP3 STEP4)
(HOLDS (BEFORE STEP3) STEP4)

(LINK STEP3 STEP0 (ALIVE ENEMYALPHA))
(HOLDS (BETWEEN STEP0 STEP3) (ALIVE ENEMYALPHA))

Figure 18 Description of the story world plan added to the knowledge base (simplified for readability)

Once the description of the story world plan is created, additonal information about each step is added to the knowledge base for the camera planner by hand. As described earlier, this information is indicates the state of characters, mood and tempo of the story at different times relative to story world actions. The information is used by the camera planner for selection of primitive shots and episode decomposition. In our example, the additional information that is added for the step shown in Figure 18 is

((HOLDS (DURING STEP3) (TEMPO LOW))

(HOLDS (DURING STEP3) (MOOD NEUTRAL))

(CLASS (ACTION GO) MOVEMENT))

(CLASS (ACTION SHOOT) 2-CONFLICT))

Goals for the camera planner are specified in the form of beliefs of the user that must hold once the story has been communicated. For this example, the goals that are specified are:

(BEL V (BEFORE[2] STEP6) (HAS DISK RYAN))

(BEL V (BEFORE STEP6) (AT RYAN MOUNTAINS))

[2] Here BEFORE and AFTER are time points that indicate the time at the beginning and the end of actions, and not time intervals relative to two actions.

Initially, the camera planner picks an open condition and queries the initial state knowledge base to find the storyworld step whose effect establishes the condition, using (EFFECT? ?step (HAS DISK RYAN)). The step returned is STEP4 : (Pickup Disk Ryan). As a result, a (LONG-SHOT RYAN) step is added in the camera plan with assertive links that require the LONG-SHOT to execute concurrently with STEP4 (e.g., LINK<(BEGIN (at-start STEP4)), (END (at-end STEP4)). ((HAS DISK RYAN))>).

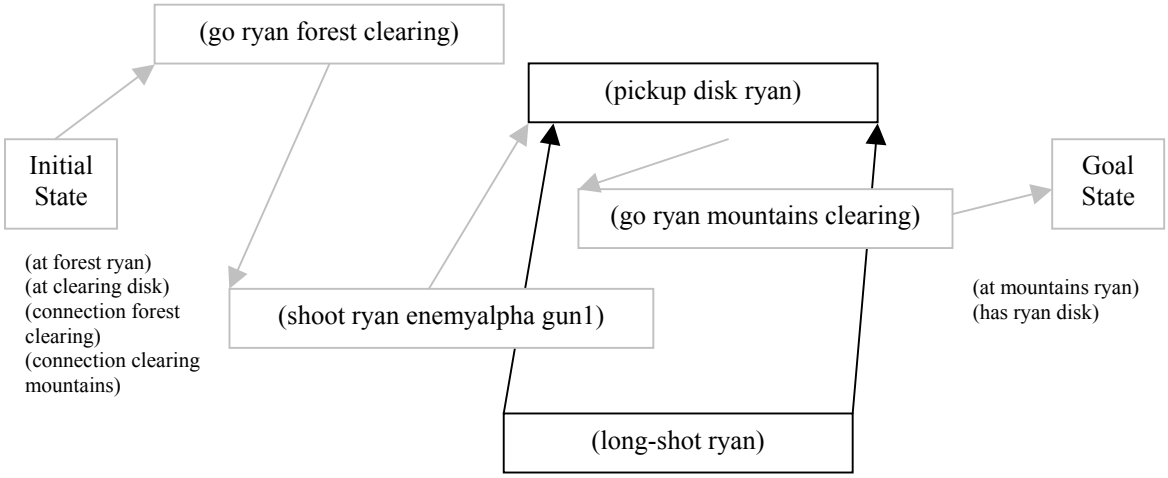

Figure 19 Adding a primitive step to the camera plan

The planner then checks the knowledge base to determine the causal link that leads in to STEP4, specifically STEP2 : (GO RYAN FOREST CLEARING). Next, the camera planner adds the abstract step (MOVEMENT RYAN FOREST CLEARING) and expands the search space into primitive actions (TRACK RYAN FOREST CLEARING) for one branch and two shots establishing (AT FOREST RYAN) (AT CLEARING RYAN) (LONG-SHOT1 RYAN) (LONG-SHOT2 RYAN).

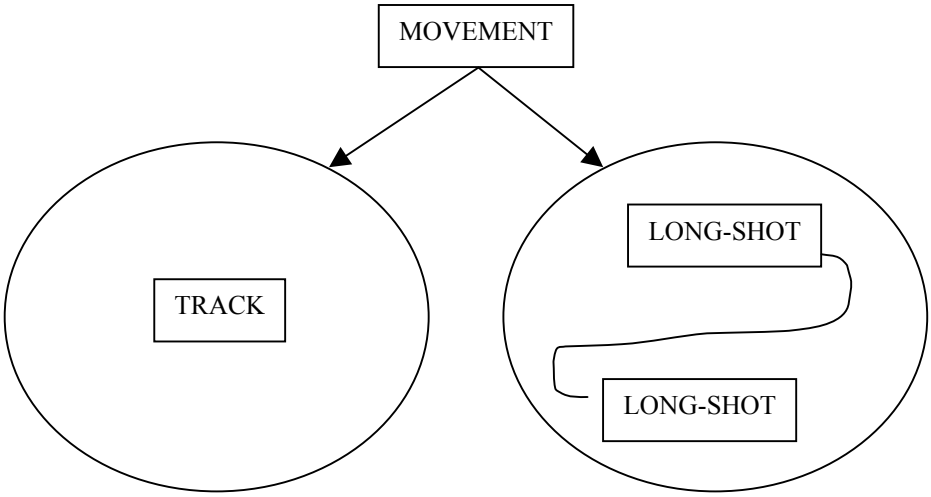

Figure 20 Expansion of Movement Idiom into continuous movement and two cuts establishing the initial and final locations

This process is continued until all the open conditions have been satisfied by the actions in the camera plan. At the beginning of the story the characters and objects are established. The resulting plan is illustrated in Figure 21.

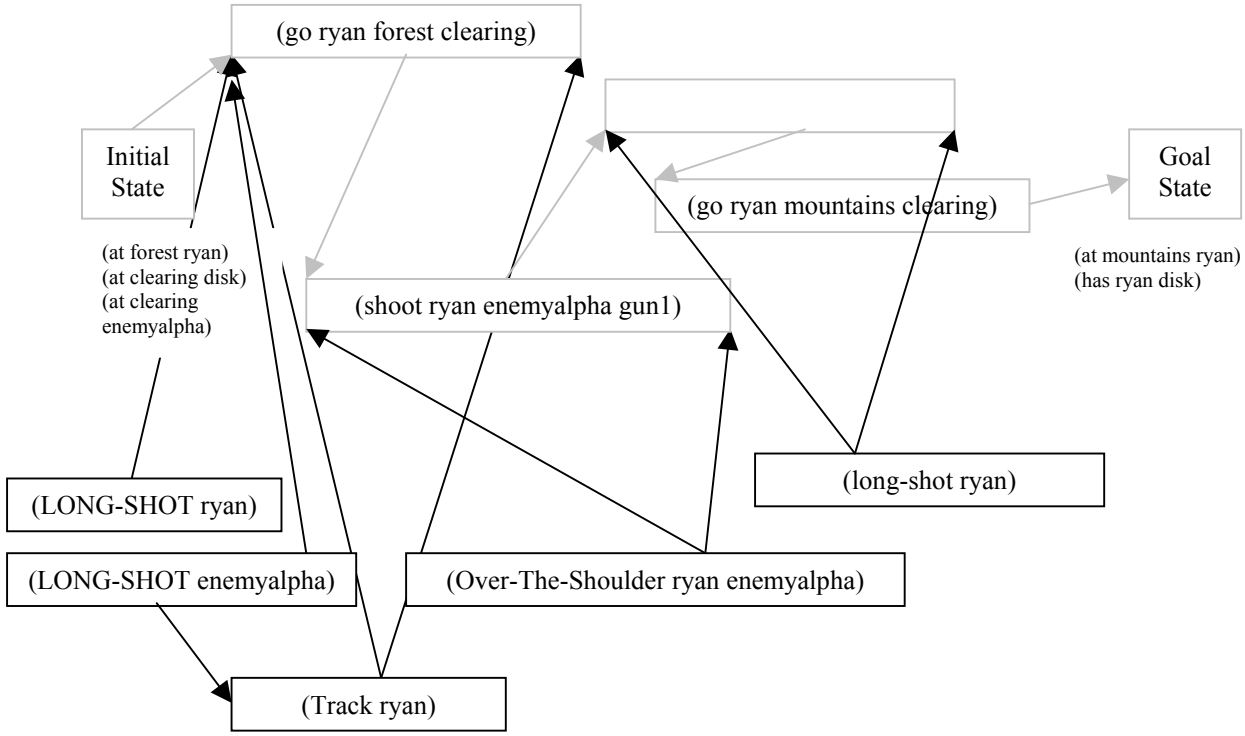

Figure 21 Relationship between camera plan and storyworld plan

This plan is realized in the Unreal Tournament 2003 game engine as shown in the screenshots for Figure 22 and Figure 23.

Figure 22 Tracking shot of Ryan running through the forest

Figure 23 Long-Shot of Enemy with Mountains in the background

6. Conclusions

The Mimesis FILM system is in its early stages of development, both theoretically and practically. In this thesis, I have focused on developing an initial correspondence between discourse planning techniques and the generation of effective cinematic shots and shot sequences. The implementation that accompanies this work indicates that the model of cinematic discourse can be used to generate shot sequences that, while underconstrained with respect to the full demands of automatic camera control, provide an effective means for viewing unfolding plot.

5.1 The Good

Through the implementation of the Mimesis FILM system I have demonstrated the application of conventional discourse models, primarily used in natural language generation, to generating specifications for controlling a camera in a 3D virtual environment, thereby creating a cinematic visual discourse of a dynamic narrative structure. This approach motivates a more formal study of the rhetoric structure conveyed by the visual medium.

I have formalized the cinematic idioms as abstract communicative plan operators that model the intentions of the cinematographer in selection of shots. The idioms are defined as sequences of primitive camera actions that could be implemented by any underlying graphics system. I have provided one such implementation of the camera primitives using the Unreal Tournament game engine. This work complements the geometric constraint solving approaches described in section 2.1.3.

The work presented in this thesis is more grounded in communicative principles from a theoretical standpoint than the previous work in this area. Rather than focusing on geometric optimization of shots or scripting of idioms, I have modeled the cinematographers intentions in generation of coherent sequences of shots.

5.2 The Bad and the Ugly

The system is not complete; it is at a stage where it is capable of generating and playing out sequences of shots, but due to the limited idiom database it is not as expressive as a human cinematographer would be. Thus, it would not be fair to evaluate the system based on its expressivity, but rather on its potential as it is based on well-established theoretical grounds. Evaluation of this system is also difficult due to the lack of a theory of discourse structure for cinematic or visual media. An approach similar to text-based corpora analysis applied to a visual medium could lead to a more expressive system based on empirically derived rules rather the use of the current operators developed by incorporating the idioms that are informally adopted standards in cinematography.

7. Future Work

The end is nigh,
So is a new beginning.
-Arnav

This thesis proposes a novel way of solving the problem of virtual cinematography, especially in story driven virtual environments. In this work, however, we have only addressed the cinematic presentation of stories generated by a narrative planner. For a general application of this solution, our work will be adapted to the three main types of virtual environments that are story based.

1. Pre-scripted storylines with computer mediation/intervention
2. Pre-scripted branching storylines with dynamic branch selection
3. Dynamic computer generated storylines

Unlike the theory of speech acts that is driven by theories of discourse structure in natural language text, this work motivates the analysis of film for identification of rhetorical structure in the grammar of the film language. Some initial efforts have already been made for such analysis[6]. Current narrative planners generate action sequences for description of a story. More expressive story planners that model not only coherent plot lines, but also encode the stylistic information and stage cues that are embedded in a traditional hollywood script. A number of factors like Camera, Lighting, Sound effects etc. are responsible for creating a rich narrative experience for the user. Research in these areas have reached a level of maturity where there is an increasing need for a general ontology for interactive narratives[20].

Anime is a very popular form of cinematic expression in Japan and has a wide audience all over the world. There is a rich vocabulary used by anime developers especially considering the limitations of the medium.

Figure 24 : Anime genre (cowboy bebop series)

Since the graphical virtual worlds do not bind the developers with physical constraints for camera positions, incorporating some of the techniques from this genre could prove to be very effective for an automated camera planning system for virtual worlds.

8. LIST OF REFERENCES

1. J. F. Allen. *Maintaining knowledge about temporal intervals*. Communications of the ACM, 26(11):832-- 843, 1983.
2. Amerson Dan, Kime Shaun, *Real-Time Cinematic Camera Control For Interactive Narratives*, Working notes of AAAI Spring Symposium, Stanford, CA 2000
3. Arijon, Daniel *Grammar of Film Language*, Los Angeles, Silman-James Press, 1976
4. Bares William, Lester James, *Cinematographic User Models for Automated Realtime Camera Control in Dynamic 3D Environments*, Proceedings of Sixth International Conference, UM-97
5. Branigan E, *Narrative Comprehension and Film*, London and New York, Routledge, 1992
6. Brett Adams, Chitra Dorai and Svetha Venkatesh, "Towards Automatic Extraction of Expressive Elements from Motion Pictures: Tempo," Special Issue on Multimedia Database, *IEEE Transactions on Multimedia*, 2002.
7. Chatman S, *Story and Discourse: Narrative Structure in Fiction and Film,* Ithaca London, Cornell University Press
8. Chatman S, *Reading Narrative Fiction*, New York, Macmillan Publishing Company
9. Christianson David, Anderson Sean, He Li-wei, Salesin David, Weld Daniel, Cohen Michael, *Declarative Camera Control for Automatic Cinematography*, Proceedings of AAAI-96
10. Drucker Steven, Zelter David *Intelligent Camera Control in a Virtual Environment*, Graphics Interfaces 1997
11. Drucker Steven *Intelligent Camera Control for Graphical Environments*, PhD Dissertation, MIT 1994
12. M. Elhadad and J. Robin. 1998. *Surge: A Comprehensive Plug-in Syntactic Realization Component for Text Generation,* Computational Linguistics Volume 99, Number 4
13. M. Elhadad. 1993. *Using Argumentation to Control Lexical Choice: A Functional Unificatio Based Approach*. Ph.D. thesis, Columbia University.
14. Ferguson, George *Explicit Representation of Events, Actions, and Plans for Assumption-Based Plan Reasoning*, Technical Report 428, Department of Computer Science, University of Rochester, Rochester, NY, June 1992.
15. Grosz Barbara, Sidner Candace, *Attention, Intention and Structure of Discourse*, Proceedings of ACL 1986
16. Grosz Barbara, Pollack Martha, Sidner Candace, *Discourse,* Foundations of Cognitive Science MIT Press-M. Posner Ed. 1989

17. He Li-wei, Cohen Michael, Salesin David *The Virtual Cinematographer-A Paradigm for Automatic Real-Time Camera Control and Directing*, Computer Graphics Proceedings, 1996

18. Hovy, E. *Automated Discourse Generation Using Discourse Structure Relations*. Artificial Intelligence 63, pp. 341-385, 1993

19. Hovy E., *Pragmatics and natural language generation*, Artificial Intelligence 43(2), pp 153--197, 1990

20. Magy Seif El-Nasr*, Story Visualization Techniques for Interactive Drama,* Proc. of AAAI Spring Symposium, CA, 2002.

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

22. Maybury M, *Communicative acts for explanation generation*, International Journal for Man-Machine Studies (1992) 37, 135-172

23. W. C. Mann and S. **A**. Thompson. *Rhetorical Structure Theory: A Theory of Text Organization*. Technical Report, isi Reprint Series isi/RS-87- 190, USC Information Sciences Institute, Marina Del Rey, Ca., June 1987

24. McDermott Scott, Li Junwei, Bares William, *Storyboard Frame Editing for Cinematic Composition*, Proceedings of IUI-2002

25. Monaco James, *How To Read A Film*, New York, Oxford University Press, 1981

26. Moore, J.D. & Paris, C.L. (1989). *Planning text for advisory dialogues*. In Proceedings of the 27 th Annual Meeting of the Association for Computational Linguistics, pages 203--211, Vancouver, B.C., Canada.

27. S. Kambhampati, C. Knoblock and Q. Yang. *Planning as Refinement Search: A Unified framework for evaluating design tradeoffs in partial order planning*. Artificial Intelligence special issue on Planning and Scheduling

28. Thompson Roy, *Grammar of the Edit*, Media Manual Series, Redwood Books, 1994

29. Tomlinson Bill, Blumeberg Bruce, Nain Delphine, *Expressive Autonomous Cinematography for Interactive Virtual Environments* Fourth International Conference on Autonomous Agents, Barcelona, Spain 2000.

30. R Michael Young, *An Overview of the Mimesis Architecture: Integrating Intelligent Narrative Control into an Existing Gaming Environment,* The Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, Stanford, CA, March 2001

31. Young, R. M., J. D. Moore, and M. E. Pollack (1994). *Towards a principled representation for discourse plans*. In Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, Hillsdale, New Jersey, pp. 946--951. Lawrence Erlbaum Associates.

32. Young R Michael, Moore Johanna, *DPOCL: A Principled Approach To Discourse Planning*, Proceedings of the Seventh International Workshop on Text Generation, Kennebunkport, ME, 1994

33. Young, R.M.: *Creating Interactive Narrative Structures: The Potential for AI Approaches*. AAAI Spring Symposium on Artificial Intelligence and Computer Games, AAAI Press (2000).

34. Young, R. M., M. E. Pollack, and J. D. Moore (1994). *Decomposition and causality in partial order planning*. In Proceedings of the Second International Conference on Artificial Intelligence and Planning Systems, Menlo Park, CA, pp. 188--193. AAAI Press.