Narrative Guidance of Interactivity

by

Tinsley Azariah Galyean III

B.S. University of Colorado, Boulder (1988)
Sc.M. Brown University (1991)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 1995

Author ____
Program in Media Arts and Sciences
May 5, 1995

Certified by ____
Glorianna Davenport
Associate Professor of Media Technology
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by ____
Stephen A. Benton
Chairman
Department Committee on Graduate Students
Program in Media Arts and Sciences

Narrative Guidance of Interactivity

by

Tinsley Azariah Galyean III

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
on May 5th, 1995
in partial fulfillment of the requirement for the degree of

Doctor of Philosophy

Abstract

Making narrative *interactive* promises to add a new depth and richness to the act of storytelling. It will allow us to experience story at a new level, more profoundly affecting us than ever before. But to do this we need to understand how a viewer can participate in a drama in interesting and engaging ways without disrupting the plot -- which is the essential structure that transforms a mere sequence of events into a *story*.

This thesi~~s describes an approach to interactive narrative that di~~vides narrative into levels; the plot level which represents the high level goals, intentions, and events of the story, and the presentation level representing the geometry, motion and camera which produces the images seen by the viewer. Today's immersive interface technology provides a seamless and compelling link between the viewer and the presentation level. But the link between the plot level and the presentation level remains unexplored. This document describes techniques as well as a theory for seamless integration of transitions (the plot's influence on the camera) and the manipulation of staging (the plot's influence on the geometry and motion) into interactive, immersive narratives.

By introducing these techniques while allowing the viewer to influence the presentation, a new method and vocabulary for storytelling has been created. This new partnership between the story and the viewer allows the presentation to be manipulated while the plot assures that story will find the viewer regardless of his/her actions.

Thesis Supervisor: Glorianna Davenport
Title: Associate Professor of Media Technology

This work was sponsored in part by the members of the Television of Tomorrow Consortium.

Committee

Chairperson ____________________

Glorianna Davenport
Associate Professor of Media Technology
Program in Media Arts and Sciences

Member ____________________

David Zeltzer, Ph.D.
Principal Research Scientist
Research Laboratory of Electronics

Member ____________________

Professor Larry Friedlander
Department of English
Stanford University

Dedication

to Sheri and Tinsley Jane

Acknowledgments

It goes without saying that my committee deserves many thanks. Glorianna Davenport for her guidance and for providing the environment in which this work could be done. David Zeltzer for supporting my work from the beginning of my time at the Media Lab. And Larry Friedlander for his insightful expertise and nurturing personality.

The Virtual Reality Laboratory exhibit (chapter seven) was developed by Art Technology Group (ATG) for the Chicago Museum of Science and Industry. Thanks to Joe Chung and Jeet Singh I was given the opportunity to direct this piece. A project of this size requires the help of a number of people. In particular: Martin Friedmann, Stephen Clark, Andy Schwarz, Keith Baccki, Andy Hong, and David Rose. This project was partially funded by the NSF, Chicago Community Trust, Howard Hughes Medical Institute, and the Radiological Society of North America. Modeling software was donated by Software Systems and SoftImage. 3D Datasets were donated by Viewpoint. 2D textures were donated by Form and Function.

Dogmatic (chapter eight) was developed at the MIT Media Lab of course with the help of others. In particular, Bruce Blumberg, with whom the directable characters work (chapter eleven & appendix A) would not have been possible. Thanks to his insights, professionalism, and sense of humor, a great system and friendship were built. Teresa Marrin, not only for her original compositions but also her infectious enthusiasm. Taylor Galyean and Marilyn Feldmeier for applying their creative energy to the generation of the models that give *Dogmatic* its unique look. Damon Horowitz for lending the technical support needed to bring music to *Dogmatic*. Viewpoint and Virtual Research for contributing data and equipment.

There are a number of other friends and colleges whose conversation has been inspirational to this work. In particular Steve Drucker, John Hughes, Mark Lucente, Shawn Becker, and the members of the interactive cinema group: David Kung, Kevin Brooks, David Tames, Lee Morgenroth, and Natalia Tsarkova.

I cannot say thank you enough to Sheri Galyean whose contributions are endless. Besides her tireless emotional support, she is responsible for the *Dogmatic* story, contributing key ideas, and reading my writings more than anyone should ever have to.

Contents

Chapter One

Chapter Two

Chapter Three

Chapter Four

Chapter Six

A Taxonomy 77

Chapter Seven

First Experiment 93

Chapter Eight

Dogmatic 101

Chapter Nine

A Computational Plot Structure 107

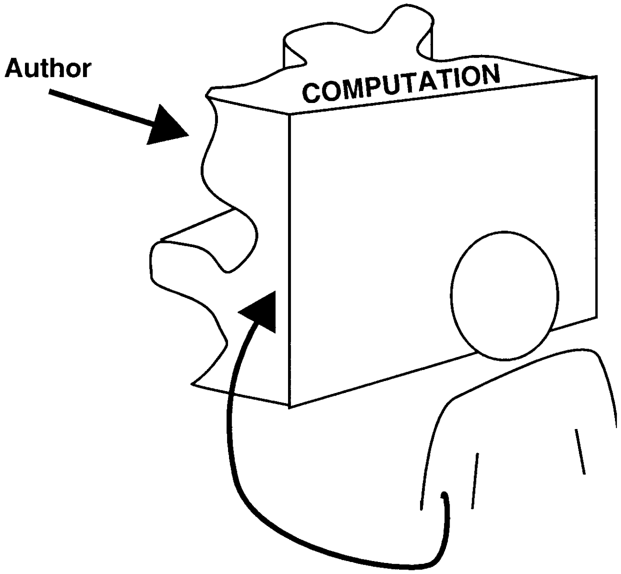
Chapter Twelve

Chapter One

Introduction

Interactivity is about freedom — the freedom to influence, to change, and to experience situations that might otherwise be impossible. It is this freedom, coupled with an unprecedented sense of presence, that immersive interface technology provides. This technology promises to take us anywhere and to allow us to do anything. A person can be placed in synthetic environments limited only by the author's ability to conceive and construct. The user's movements can be smoothly and continuously monitored and interpreted, giving his/her a kinesthetic link to the virtual environment. This link allows the action of interacting to drop from consciousness. Users can concentrate on what they are experiencing and not on how they are experiencing it. Actions are immediately converted into results in the virtual world.

Just as this freedom makes immersive interfaces compelling, structure or plot makes narrative compelling. Authors construct and con-



In traditional narrative the author has control over all material up to the point it is presented to the audience. When story becomes interactive a computational model becomes part of the story representation. The author is then responsible for defining this computational terrain and what handles into it the audience will have.

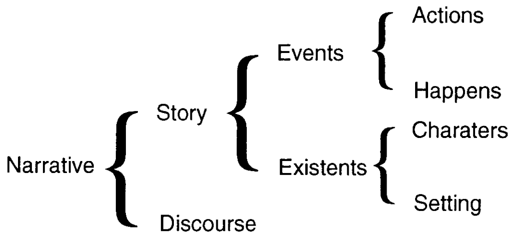
strain the presentation of a story in order to more profoundly affect their audience. All elements are orchestrated toward this effect. Characters and their actions are carefully constructed and the time line along which events are revealed to the audience is deliberately crafted. The story is both presented to and imposed on the audience, and events are structured to produce a tight storytelling. A narrative's effect on us is a product of both the material and a how it is delivered, i.e. the sense of flow that is created by the presentation. This highly structured art form might seem to be at odds with the freedom of interaction provided by immersive interfaces.

The ability to smoothly and continuously interact adds a new depth and richness to the material being presented. A stronger and more compelling connection between the user and material is created. The promise is that by bringing interactivity to storytelling an author may better reach his/her audience and more profoundly affect them. For this merger to happen a new vocabulary for storytelling will need to emerge. This new art form will give us new ways to learn and to be entertained. To successfully merge the art of storytelling and the ability to interact, we need to bring structure to the freedom interactivity gives us. We need to find ways to guide the user's interactions while permitting story flow to continue. But to maintain the sense of freedom, these techniques must be transparent to the user. In concert with their free actions, users must be smoothly steered through the plot points as the author intends. If this guidance can happen seamlessly, a sense of freedom will be maintained while a coherent story is told. Users will this freedom while the system assures that the story gets told. This work presents new techniques for this guidance. We work to direct the user's attention, to manipulate the user's point of view, and to orchestrate the transitions necessary to weave a compelling story.

Immersive interfaces provide a great sense of presence and freedom while narration or narrative structure allows an author to weave a engaging presentation. The hope is, that by bringing together the best of these compelling experiences, a new type of experience can be made and that this new experience may even be more compelling or at least more appropriate for certain applications. To achieve this goal however a new representation for narrative must be developed and this new representation will have to include computational elements.

Narration

The dictionary defines a narrative as "an account of incidents or events." This process of describing a sequence of fictional or actual events is one of the main ways that we share information and experiences. Over the centuries many different techniques and media have been developed to produce narratives. These range from the first spoken words of the storytellers, to the films of modern cinema. Narrative is a part of every experience. We do not encounter narrative solely in novels, film and conversation, but in our every day activities as well. We wonder about events. We think about what could happen next. In other words, we tell ourselves small stories. And narrative is one of the most powerful ways in which we do this. Edward Branigan, begins his book *Narrative Comprehension and Film* (Branigan 1992), with a similar argument stating that narrative is not just a means of communication but is also "a fundamental way of organizing data." He introduces this argument by explaining that it is important to note that a narrative is both the events that make a story and the process by which those events are presented. Likewise, Chatman (Chatman 1978) divides narrative into two parts: story and discourse. Story is the content, consisting of both



Chatman's diagram of the narrative elements, most importantly dividing narrative into the *what* (story) and the *how* (discourse).

events and existents (characters and setting). The discourse of the narrative is the expression of the story. The means by which the content is communicated. In other words, he talks of the story as the *what* and the discourse as the *how.* Chatman is not the first to make this distinction, but I have chosen his discussion for clarity.

This discourse or the *how* is the narration of the story. When I use the word narration, I, like Branigan, use it in its broadest sense. The narration is all those elements and techniques that are used to control the flow of information from the author to the audience. This means that narration includes not just a narrator's voice but also the staging of a scene or the editing of a film. For the process of storytelling to become interactive the narration must become adjustable. The new found freedom given to an audience via immersive interaction will allow them to change what is presented. To maintain the story's integrity while the audience is interacting, new narrative representations need to be developed. These new representations will not only need to accommodate change imparted from the interactive audience, but will also be able to adaptivily change the narration itself to achieve the story goals (plot points) in spite of the audience's actions.

Narrative representations

A narrative can be represented in many different ways. Below are a few examples of these different representations. A film is a sequence of frames (images) that are shown at a constant rate (24 per second). In a film the narrative representation, at its lowest level, is nothing more that a sequence of images. The written narrative can take a number of different shapes including the novel, the screenplay, or just a stack of letters. But in each of these cases the lowest level of representation is the words (or letters) on the page. Here, both the words and their positions on the pages together form the representation. In the oral tradition the story is converted from its representation in the mind of the storyteller to a sequence of vocal utterances. What is interesting to note in this example is that there are two representations. The one in the storyteller's mind, and the one revealed to the audience. The representation in the mind of the storyteller has the advantage of being flexible. It can be adjusted to the needs and situation of the current telling. But it is not in a form that can be given directly to another person. It must be interpreted, and a low level verbal representation must be reconstructed.

Today's technology can be used to provide alternative representations. A film has many different elements: the setting, characters, actions and happenings. These elements may take many different forms during the production process but the final representation and the only representation that is preserved is the frames. This greatly restricts the potential for interaction. With the advent of fast computational hardware and real-time computer graphics, these elements need not be stored as frames. The setting and character can be represented (stored) as three dimensional models. There motion and the camera positions can be preserved in a scripting language. With this information, the computational hardware can reconstruct the frame through the process known as *rendering*. Like the representation in the mind of the storyteller, this representation requires reconstruction, which in this case is done by a computer.

We can think of this reconstruction process as an extension of the presentation process. A film must be projected, an audio tape must be played, a book must be read, and in the example given here, the models must be rendered. The computer graphics representation is an example of a *higher level* representation of a narrative. It is higher level because the presentation process is more complicated. The representation is farther from a form that can be digested by an audience. When the representation is farther from a presentation, it is more costly to reproduce the presentation. But though the price of a more complicated presentation process is paid, the benefit of flexibility is added. When the presentation is being created on the fly, there is the ability to alter the details of the presentation to better suit the current needs.

Looking again at the computer graphics representation of a narrative described above, it can be seen that the camera's relationship to the scene may be scripted. This scripted information is used to position the virtual camera and reconstruct the frames. If all that is done is to routinely position the camera and then render the frame, there is little if any advantage over film. If the audience is able to influence (through input) the way the camera presents the narrative or what action a character takes, this representation can accommodate the viewer's preferences. In this example, the reconstruction or presentation process becomes an interactive one. This is only one of many examples of how higher level representations of narrative can provide more flexibility than the traditional representations.

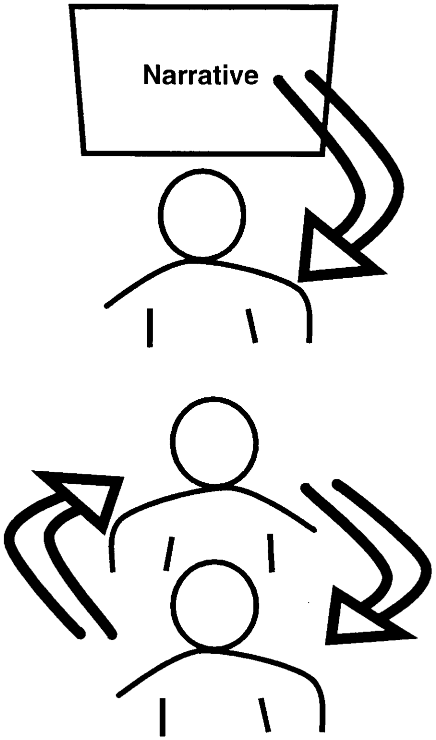
These higher level representations will be called *content-based representations*. The dictionary gives a couple of definitions for "content" pertinent to this discussion: 1) essential meaning; significance 2) the events, physical details, and information in a work of art. There are a number of applications for which a content-based representation of narrative would be useful. A few are listed below:

- Interactivity: As the example above illustrates a content-based representation can provide the flexibility to introduce audience input during the presentation process.
- Compression: In many circumstances the digital space

needed to store a narrative can be greatly compressed by using content-based representations. It takes fewer bits to store computer graphics models and a script than to store the images for each frame.

- Multiple output formats: The rendering process described above does not necessarily need to generate images. Alternatively, it could generate text or audio that describes the scene. In this way a content-based representation can be independent of the display or presentation method.
- Searching: A content-based representation contains knowledge of events and physical details. If structured correctly this knowledge can be used to search for particulars. For example, "find all the shots containing a blue chair", or "show me the part where the woman is kidnapped."
- Constructing new narrative: Content-based representations have the potential for being used as a database of events, characters, settings, etc. that can be pulled from to construct a new narrative. For example, "give me a character like an Indiana Jones but with a lust for power like the Godfather, and place him in a Blade Runneresque city"

This thesis will focus specifically on content-based representations of narrative for interactive cinema. While what is learned and developed may directly or indirectly apply to content-based representation for other applications (like those listed above), there is more than enough need for research in the issues of narrative representation for interactivity.



Top: The one-way channel of traditional narrative.

Bottom: The two channels that make conversation interactive.

Interactivity & narrative

Traditionally movies deliver a narrative to the viewers through a one-way channel. Information flows from the film or video to us, but no information flows from the audience back to the movie. We have no control over what events are presented or how those events are presented. The only control/input we have is whether or not to go to the movie. However, by sacrificing input, we as the viewers are taken on a journey. This journey carries us through a set of events, giving us an experience we may never otherwise have had, and teaching us something that we might otherwise never have learned.

Alternatively, the experiences of our lives are always interactive. Information flows from our environment to us while we impart action and information into our environment. This is truly interactive; a two way channel is opened. Except in rare situations, most of our life experiences do not tightly weave a narrative that is at every turn as compelling and engrossing as a good movie. The question is then, what lies between these two extremes? Can a life-like experience be given narrative structure? Can a movie be interactive?

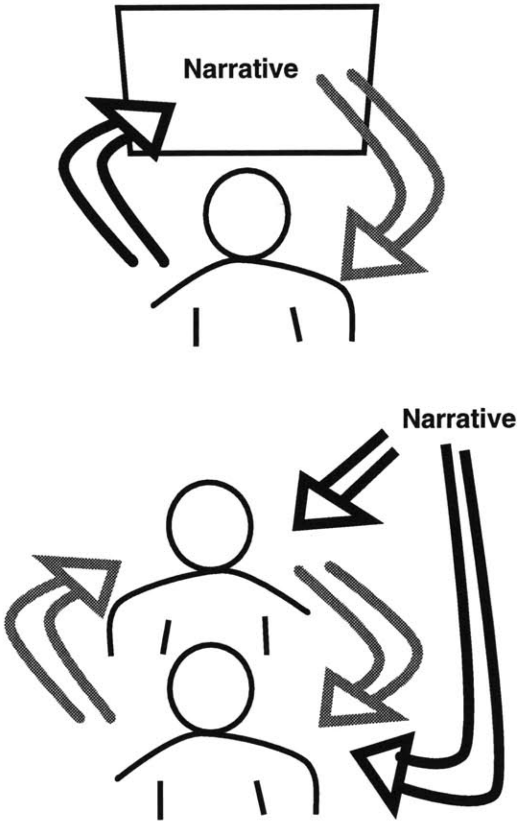
People have speculated about these questions for a number of years now, but it is only in the last decade that technology has begun to provide the tools to create interactive narrative experiences. These experiences range from environments open for exploration (like Multi-User Dungeons and Virtual Environments) to more structured experiences (like many CD-ROM based interactive movies and video games.)

It is natural to look at the issue of creating an interactive narrative in one of two ways by adding interactivity to a traditional narrative structured like movies, or by adding narrative content to an already highly interactive environment. Both of these approaches imply that interactive narrative is the product of gluing together a narrative structure with an interactive method. But these approaches to interactive narrative fail to illustrate that traditional narrative representations do not accommodate interactive input from the viewer. In other words, it is necessary to develop a new narrative representation.

The general problem

Narrative, or story in general, plays a significant role in our lives. Narrative structure is at the core of our communication — how we organize information — and is essential to much of our entertainment. It is the key element in most successful books, films, and verbal discourse. Bran Ferrin, from Disney's Imagineering, stated in a talk that 90% of what Disney does is "story," the implication being that story is greatly responsible for Disney's success. It won't matter, according to Ferrin, how clear the image is, how high the resolution is, or how great the sound is, if the story (narrative) is not captivating (Ferrin 1994).

Interactive immersion gives the users an unprecedented sense of presence. The user feels more a part of the material being presented than ever before. As this technology evolves it will constitute a new medium and with it a new art form. We, the audience, will enter fantasy worlds, no longer just passively observing. We will interact with these worlds and actively alter them. But this technology alone does not tell us a story. The promise of "virtual reality" technology is that it will allow us to live any experience we desire, in other words, to "go anywhere and do anything." (Bates 1992) But to do this we need to learn how to use it more effectively in order to create and experience stories.



Top: Adding a channel from the viewer to the narrative to make it interactive.

Bottom: Adding narrative to an already interactive conversation.

Making narrative interactive promises to add a new depth and richness to the act of storytelling which will allow authors to better reach their audiences. It will allow us to experience story at a new level, more profoundly affecting us than ever before. To do this means overcoming the problems of how the viewer's (user's) interaction relates to story. Solving these problems means creating a partnership between the story and the viewer. This relationship allows the narrative to be personalized by the viewer's actions. Every contribution that helps us understand, categorize, and manage this relationship between the viewer's interaction and the narrative will help us find a solution to these problems. By finding a solution we hope to create a new vocabulary for telling story and to discover new ways to learn, which could significantly change the nature of entertainment.

Emergence vs. guidance

To date most interactive narratives have put the emphasis on the word "interactive." In other words, asking the question "How can interactivity empower the user to influence his or her experience?" This has meant giving the user control to construct the narrative by providing the freedom to steer and the ability to influence how the narrative space is navigated. However, there is an alternative approach. That is to ask the questions "How can Interactivity be employed by the author to better tell his/her story?" and "How can the narrative be used to guide the interaction of the user?" In this approach the story environment is manipulated to ensure that the user experiences the narrative that the author intends. I call this "Narrative Guidance of Interactivity."

Narrative emergence from interactivity

We all construct narratives out of our daily activities to help us remember, understand, categorize and share experiences. It is this skill that many interactive systems exploit. They give us environments to explore. We, by combining the elements of these spaces with our goals (the user's goals), allow a narrative to emerge. If any narrative structure (or story) emerges it is a product of our interactions and goals as we navigate the experience. I call this "Emergent Narrative." This approach has provided a number of successful interactive experiences such as flight simulators, games (i.e. DOOM), and narrative puzzles like MIST.

Narrative guidance of interactivity

The alternative approach is the process of empowering the author to bring structure to the experience, which makes this medium more appropriate for a wider range of applications. Introducing narrative structure will increase the ability to use this technology for teaching, storytelling, advertising and information presentation. But to incorporate narrative structure into an experience, we will need to balance the interaction (exploration) with an ability to guide the user, while at the same time maintaining a sense of pacing or flow through the experience.

In the narrative guidance model the presentation is manipulated to assure that the user will be told the story regardless of their interaction. In other words, the story remains the same at a high level while the presentation of the story varies. In addition, the narrative guidance model uses its ability to manipulate the presentation to control the flow or pacing of the story.

Take, for example, the following scenario generated in a virtual world. You are a character in a story and are standing beside a road. The camera starts in your body functioning as your eyes. As you look around at the setting, you hear a car approaching in the distance and you take the opportunity to stick out your thumb and hitchhike. When you turn to see the oncoming car, the camera cuts to a close-up of the person in the car. This gives you more information about the approaching driver. The camera cuts back into your body and the car pulls over. You begin to run toward the car, the camera cuts to a distant shot showing you coming to a stop beside the car (time is foreshortened). Cutting back you look down toward the door handle. The camera cuts to a close-up as your hand opens the door. As you get in the car, you pan the inside of the car and a close-up of a gun between the seats is inter-cut. This sequence gives an example of how the presentation is manipulated to both emphasize particular elements and to control time and pacing. The viewer is nudged through the narrative while maintaining the perception that anything can happen.

Assume that getting the viewer into the car is necessary for this story. In fact it is a plot goal. The goal was satisfied in the scenario given, but what if the viewer had not chosen to hitchhike? The solution is for the system to adapt the staging of the scene to achieve the goal. For example, the car might pull over directly in front of you and the driver would wave a gun demanding you get in the car. The car might screech to a halt hitting you (knocking you unconscious) and as you wake up, you find yourself in the car. The point being, plot level guidance can manipulate you as a viewer while providing the freedom to let your actions adjust the presentation. By manipulating the story world and how it is revealed to the viewer, the plot can provide the viewer with the illusion of having an infinite number of options even if the system is only prepared with a few. The final result is that the viewer, at the presentation level, has created his/her own experience while at the plot level the viewer has experienced the story the author intended. The computational system accomplishes this by dynamically directing the presentation of the story events. These directives are given to both characters and to the camera.

Contributions & thesis organization

In the previous section I have outlined a philosophy for approaching interactive storytelling systems. It is in the context of this approach that this thesis makes its contributions. These include:

- Four principles or guidelines for approaching the construction of an system that will provide narrative guidance of the interactivity. They are given at the beginning of the next chapter (the approach chapter).
- The approach chapter then articulates the pieces needed to execute this approach in the context of an immersive, three dimensional environment. The thesis later focuses on three of these pieces, devoting a chapter to each: computational plot, directable characters, and "cutting" in an immersive environment.
- An algorithm for computationally representing plot is presented. The work focuses on enabling the author to construct a plot that can be both responsive to the user while also ensuring the story is told.
- An architecture for representing "directable characters" is presented. This architecture is flexible enough to make it

easy to add new abilities to existing characters as well as to create new characters.

- The issues behind introducting real-time editing (in particularly "cuts") into immersive environments is explored. The general problem is articulated and particular examples are illustrated. These examples serve to show that cuts can successfully be used in immersive environments.
- The thesis also presents an early experiment and a final interactive immersive piece that illustrate the successful uses of these techniques and algorithms.

In addition to these contributions several additional chapters are included. One presents a new analogy for how narrative spaces should be interactively navigated. Another gives a more complete discussion of how this architectural representation of narrative was developed. A third chapter presents a taxonomy of interactive narrative that helps to position my work with respect to other systems.

Chapter Two

Approach

I separate narrative into multiple levels of representation:

- plot level -- representing the high level goals, intentions, and events of the story.
- presentation level -- representing the geometry, motion and camera which produce the images seen by the viewer.
- viewer level -- which includes the visual, audio, tactile, etc. elements used to convey the story to the viewer.

First I list four principle for narrative guidance of interactivity. The rest of the chapter argues the value of viewer interaction at the presentation level, and shows how immersive interface technology and content-based representations provide this by linking the viewer level with the presentation level. Next, the use of transitions and staging is identified as a means to link the plot level with the presentation level.

Principles for narrative guidance of interactivity

Here I outline four qualities that I think are important to the development of narrative guidance. These have guided my research work.

- **Temporal Structure.** Interactive narrative should have temporal continuity. Traditionally the events presented in a narrative world have a structured relationship to each other. A good storyteller strategically controls the time and space of the presentation to weave a tight and compelling experience. The process of narrative guidance should also do this. By allowing a narrative to guide the interaction it can be ensured that important plot points are reached while maintaining a sense of flow or pacing in the presentation.
- **Continuous Interaction.** Interactive storytelling should not pole the user for interaction. The presentation of the narrative should not stop and wait to be started by the actions of the viewer. The experience should proceed smoothly. Therefore the user's interaction should be a smooth and continuous stream of input that influences the story world, much as a rudder steers a boat. Unlike the start and stop of an system that interrupts the story awaiting input, this type of input can coexist with the temporal structure mentioned above.
- **Two Levels of Representation.** There should be two layers that make up an interactive narrative: the plot level and the presentation level. The plot level embodies the tempo-

ral structure. It manages the plot points or task level story goals that are to be attained. The presentation level is indicative of the viewer's world--the world in which continuous interaction happens. Because of this separation both continuous interaction and plot can be accommodated. It is the coupling between these two levels that allows the plot to orchestrate the events while the user's input continuously influences the presentation.

- **Shifting the Viewer's Story View.** For the plot to be able to attain a goal it must have methods for shifting the viewer's position with respect to the narrative world. In cinema this is traditionally done by transitions (e.g. wide shot to close-up, 1st person point of view to 3rd person) and staging of the character's actions. These transitions manipulate time and space to provide the link between the plot and presentation. Traditional immersive environments have not used these techniques, they have no methods for shifting the viewer. They should, and this is the focus of this research.

When these principles are achieved, the viewer has the illusion that anything can happen, while the system can constantly manipulate events to tell the story. Smooth and continuous interaction with the story world can be provided while the events in this world are orchestrated to guide and steer the viewer. In other words, the system will bring the story to the viewer.

Presentation level interaction

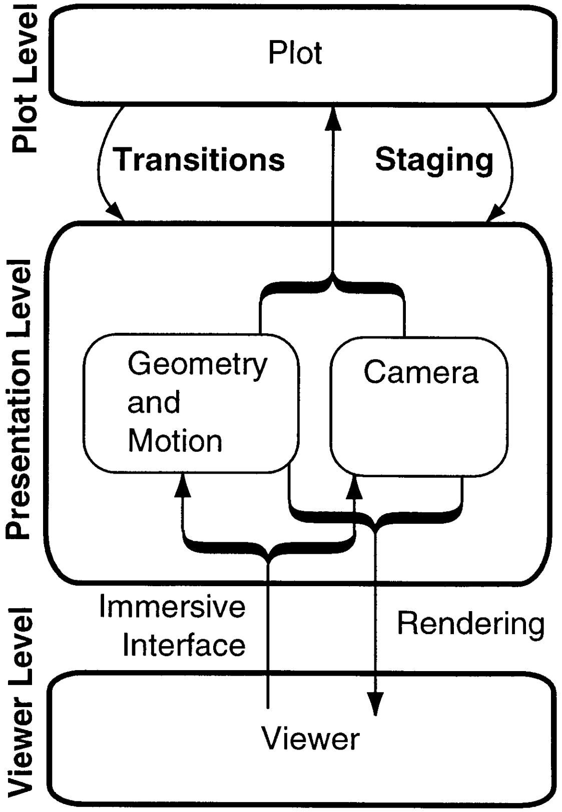
In any narrative there are many decisions to be made about how the events will be presented. In conventional film the directors and editors make these decisions during the production process. Alternatively, today's technology can provide the freedom to let these details change and adjust with viewer input. This allows the viewer to influence the details of the presentation. For example, details of the camera, character movement, or props can become free to change. Will the camera be looking at Tom or Mary? Will the character reach for the gun with her left or right hand? Is his hat red or blue? I call this level of interaction *presentation level interaction*. The presentation level includes those details that can change with different presentations of the same narrative. The presentation level interaction allows input from the viewer to be continuously incorporated into the narrative. When the viewer is continuously influencing the story, the interaction does not interrupt the presentation. Interruptions like this pull the viewer out of the story and break his/her suspension of disbelief. This presentation level interaction cannot be accommodated by the digital video approach to interactive narrative because digital video stores everything as frames. Frames are fixed and un-alterable, dictating ahead of time most of the important variables of presentation.

Content based representation

Like any narrative, a digital video narrative has many different elements: the setting, characters, actions and happenings. These elements may take many different forms during the production process but the final representation (and the only representation that is preserved) is the frames. With the advent of fast computational hardware and real-time computer graphics, these elements need not be stored as frames. The setting and character can be represented (stored) as three dimensional models. Their motion and the camera positions can be preserved in a scripting language. With this information, the computational hardware can construct the frames as needed through the process known as *rendering*. A content based representation like this frees the presentation process to be adjusted by the viewer's actions.

Immersive environments

In cinema we perceive the narrative through the camera. Therefore the camera provides the most obvious way for us to interact with the presentation of the narrative. In addition to providing interaction with the narrative and the camera, the interface should have the following properties: it should be mature enough to function reliably, it should afford continuous input from the viewer, and it should be easy for the viewer to understand and learn. Today's immersive interface technology has these properties. My definition of immersive interface is fairly broad. I include not only head-mounted displays and tracking technology but other interfaces that give a user a sense of presence in the virtual space. There are a number of games on the market that do this while using a mouse or control keys to allow the user to navigate. What is important about these interfaces is the smooth and continuous input they provide. They produce a kinesthetic link between the user and the material. This allows the user's "eye" to become the camera into the virtual world. And the *act* of interacting drops from the consciousness of the user allowing them to focus on the material being presented. The immersive environment that this technology creates provides both input and output channels that the viewer can use without being distracted from the narrative.



The approach is to represent the narrative at three levels. The user is empowered to alter the presentation via the interface, but this input only indirectly affects the plot. The plot then employs staging and transition to assure the story is told.

Manipulation of time and space

The primary way in which a storyteller weaves an engaging and compelling narrative is by manipulating space and time. In film, space and time are manipulated through the use of *staging* and *transitions*, including dissolves, wipes, fades, and most commonly cuts. Staging is the manipulation of the characters and props with respect to the viewer (or camera). This manipulation helps to control the visual field of the viewer. While it does not have the same ability to manipulate both space and time as transitions do, it is a vital tool for the filmmaker. While both staging and transitions can build clear observations, establish point of view, and strategically hide information from the viewer (to build suspense for example), cuts have the magical ability to transport the viewer in space and time and allow the filmmaker to control the pacing of the presentation to build a tightly woven and compelling narrative. Because it was the introduction of cuts (or *montage*) that signaled the birth of modern cinema, this would suggest then that the manipulation of time and space is no less important in an interactive narrative.

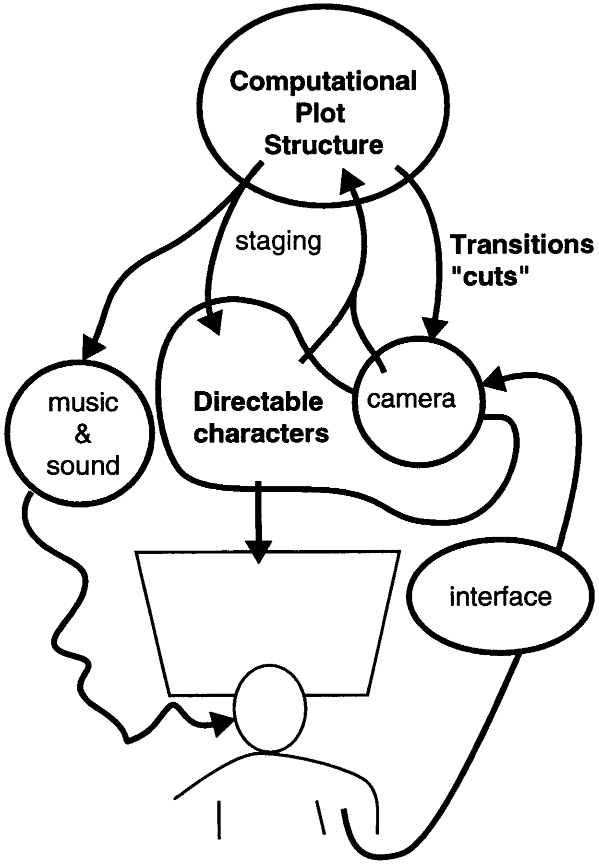
Transition and staging in immersive environments

Two of the criteria I have set forth for narrative guidance of interactivity are presentation level interaction and the ability to manipulate space and time. Immersive environments are a natural way to provide continuous presentation level interaction, and transitions coupled with staging are a proven way to allow the filmmaker to control space and time. I propose that transitions and staging be used to link the plot level representation of narrative to the presentation level. Successful incorporation of these methods for manipulating space and time into an interactive immersive environment will provide new tools with which some of today's problems with interactive narrative can be addressed.

The pieces of the puzzle

The diagram serves as an overview of the proposed approach. The three different levels of narrative representation appear at the top (plot level), middle (presentation level), and bottom (viewer level). The labeled arrows show the connections between these levels. The methods that connect the presentation level to the viewer level are well established. The process of rendering takes the geometry and camera information and generates the individual images/frames. The immersive technology provides the user with a sense of presence in the virtual environment and gives them the interface through which they can manipulate the camera and geometry. The link between plot level and presentation level is the part still undefined. To create a successful link between plot and presentation it will be necessary to find a suitable architecture.

The second diagram takes these basic pieces and breaks them down into parts that correspond to the elements of the architecture I have built. A computational representation for plot resides at the top of the architecture. This algorithm monitors the presentation level with sensors (indicated here with the dashed line). This information is then used to decide what directives will be sent back down to the presentation level. The presentation level is principally composed of a computational representation of characters. Characters are animate elements in the environment that can be given direction. By encapsuling characters as their own computational elements the staging can be accomplished by issuing directives to these characters. In this model the camera is in fact a character itself, cuts are



This figure outlines the different elements used in my approach. While all the pieces come together to create a complete experience this thesis focuses its contribution on the three areas of computational plot, directable characters, and the use of transitions or cuts.

accomplished by directing the camera to a new location and direction of view. Music and sound are also important elements of the presentation and must be modeled computationally. The interface is utilized to directly link the user to the presentation level, providing continuous interaction with the camera in particular. The bulk of the work in this thesis is in the areas of plot, characters, and the use of transitions or cuts.

So far I have maintained a separation between transitions and staging. But this distinction is less clear in a computer generated environment where there is so much freedom to manipulate the virtual world. Take for example the ability to move the sun rapidly across the sky to change the time of day or the ability to slide in mountains in the distance to change the setting. Are these changes considered transitions or are they the product of staging? It is not important to make the distinction, but it is important to develop these techniques and to learn how to select and use them to further the story. With this in mind, I use the term transitions to mean all these techniques.

How can transitions be used in an immersive environment? This question has not been researched until now. By doing so, I hope shed some light on questions like the following: How does the added complication of the viewer having a kinesthetic relationship to the camera affect the use of transitions? Can close-ups still be used to draw attention to an object or event? Is there a need to link the moment of the transition to the viewer's head movements? Can the camera cut to a new location in space and time if the head turns quickly? If you are a character in the narrative world what does it mean to have the camera cut in and out of your body? What role can sound play? Can spatially located sound clue the viewer of an impending transition? If your head movements suggest you are doing a double take and focusing on a particular object, should the camera cut to a close-up? When Eisenstein first developed his montage technique (Eisenstein 1949) his teacher Kuleshov recognized its success. To better understand how and why these cuts manipulated the audience he performed the Kuleshov's experiments (Levaco 1974). What are the equivalent experiments for today's immersive environments?

Chapter Three

Related Work

This work draws upon the background work of a number of different types. In this chapter I divide the material into six categories: narrative structure and film, human computer interaction, interactive narrative, camera control in virtual environments, action selection, and geometry and motion representations.

Narrative structure and film

There are a number of texts that present theories on narrative in general and its manifestation in film. These range from looks into basic narrative structure to theory focusing specifically on cuts in film. Here I will briefly describe a few categories for these works.

Many theorists look back at narrative, and in retrospect, decompose it into its parts. For example it is common to separate the narrative elements (character, setting, events) from the presentation (Chatman

1978; Bruner 1986; Chatman 1993). (These are not the only theorists to make these distinctions, I cite them for their clarity and my familiarity with their work.) The narrative elements represent a core of the narrative that can be presented to the viewer in a number of different ways. These elements exist, leaving it up to the author to choose the most effective presentation. It is from these theorists that I extracted the idea of multiple levels of narrative representation (i.e. presentation level and plot level).

More recent theoretical work focuses on the viewer rather than the narrative (Bordwell 1985; Branigan 1992; Bordwell 1994). These works concentrate on how viewers interpret and understand what material is presented to them. Bordwell deepens this understanding of the viewer's role by examining the viewer's activities. How does the viewer construct a narrative from what is presented to him/her. Branigan, on the other hand, draws extensively from other theorists and cognitive science to build a narrative schema. It is in this context that he develops an understanding of narration in its most general sense. For Branigan narration is the controlled distribution of knowledge. This is an important concept and directly relates to this research.

Early efforts to use cutting or montage in film mark the advent of modern cinema. Because of their development of cutting techniques, D. W. Griffith and Sergei Eisenstien are often considered the fathers of modern cinema (Monaco 1981; Cook 1990). Their writings and the writings on the now famous Kuleshov experiments all serve as sources of how these techniques where both developed and understood (Eisenstein 1949; Levaco 1974). This background seems particularly pertinent because what I am proposing can be thought of as the reinvention of these basics of cinema in the context of interactive immersive environments.

The use of transitions in film has continued to evolve since their origins. With their evolution both practitioners and theorists have helped to refine our understanding of how these techniques are both used and perceived. Some approach this subject by looking at how film manipulates space and time, often enumerating specific techniques (Isenhour 1975; Reisz and Millar 1978; Burch 1981) while others look at the cognitive and perceptual issues of motion pictures (Hochberg and Brooks 1978). Both serve to help the theorist and practitioner understand how transitions are used by filmmakers to better tell their story.

The works just mentioned begin to the bridge the gap between the theoretical and the practical. There are also a number of books written to aid the practitioner (most commonly the director) in learning the art of filmmaking (Arijon 1976; Katz 1991; Katz 1992; Richards 1992). These books contain a wealth of information about how to specify the motion of both the camera and the actors to better achieve the desired story goals. These books talk about the choreography of staging and the direction of camera positions and movements while maintaining an eye on how the results will be edited together. Although these texts are focused on the traditional art of filmmaking, they have a lot to offer the efforts of making narrative interactive.

Human computer interaction

A number of issues in human computer interface are directly related to this thesis work. Reading on the subject has not only served to influence my proposed approach but has also continued to influence this thesis work throughout its development.

User interface design has evolved into a field all its own during recent years. It is unique in that the process of designing human computer interaction is considered as much an art as a development effort. Because of this, many position papers (sometimes sermons) have been written to help shed light on this still elusive art. I tend to separate these papers into two categories, those that look back at both successful and unsuccessful efforts (Hutchins, Hollan et al. 1986; Norman 1990), and those that point out future directions (Owen 1986; Fisher 1990; Negroponte 1990). Of those that suggest new paths a few make the observation that we, as users, structure a great many of our experiences as small narratives (Laurel 1986; Don 1990; Laurel 1993). With this in mind, these researchers have suggested the addition of narrative structure to the interface. While all the papers mentioned have served to influence my thinking on this subject, the one that has most influenced my approach is Lippman's concise definition of interactivity (Anderson 1989).

Interactive narrative

There are a number of systems or pieces that have begun to address the issues of interactive narrative, by either adding interactivity to narrative material, or by adding narrative structure to already highly interactive environments. These pieces tend to fall into three categories: interaction with digital video, looks at narrative structure in immersive environments, and non-immersive interactive computer graphics environments often with simple story elements.

The most successful examples of highly interactive computer generated systems are video games. Most often video games are not immersive. Video games provide continuous and immediate interaction from the player. This direct connection between what is happening on the screen and what the player is doing, creates a captivating involvement for the player regardless of whether there is any narrative structure in the experience. While there has been little or no scholarly research to explain the success of video games there is no doubt in my mind that producers and researchers of interactive narrative can learn from their success. In the past most video games contained little or no story, but today game manufacturers have begun to link games to stories. Most often this is done by taking a story from a successful film (for example *Aladdin* or *Jurassic Park*). Although this introduces a story to the experience these games do not present a compelling narrative like the films. Instead the game uses the already understood story as structure for lacing together the stages of the game.

Recently a few products that are based on digital video technology have come to the market (for example *NightTrap* and *Voyeur*). Unlike their computer generated counterparts, they use prerecorded and stored frames. This has the net effect of restricting interaction to the shot level. In essence, shots are laced together into sequences and these sequences tell a story. Because shots are the smallest building blocks available, the interaction intermittently guides how these shots are laced together. Davenport, Smith, and Pincever give a concise discussion of these issues (Davenport, Aguierre-Smith et al. 1991). There is a lot of research credited to the Interactive Cinema Group of MIT Media Lab that pursues methods for categorizing and accessing these digital video shots so that they can be interactively combined to tell a story (Davenport 1987; Rubin 1989; Davenport, Evens et al. 1993; Evans 1993; Halliday 1993). This process is most often talked of as logging (when stored) and filtering (when retrieved). Unlike digital video, my approach is not constrained to the shots stored on disk. Without this constraint continuous interaction is possible. Although the interaction is continuous the introduction of cuts requires an understanding of what makes a "shot."

When should a cut stop one shot and start another? What shot should follow the cut? In other words, there will be a need to define a "new shot" and the lessons of interactive digital video will help.

Real-time computer graphics and the advent of immersive interface technology suggest the promise of the audience being engaged in the story in a way that has been foreshadowed only in science fiction. To date the novelty of this technology was sufficient to entertain it audiences without the need for a story. As researchers look to where this technology will lead us next, a few have taken leads from theatre and film (Laurel 1986; Smith and Bates 1989). These theories, while introducing new ways to think about the issues, are only theories and do not provide the insight needed to reach an implementation. Bates took a look at the promises of this "virtual reality" technology and identified the areas that need research if this field is to meet its artistic and entertainment expectations (Bates 1992). There have been a few immersive experiences created which strive to be more than a virtual environment open to exploration by incorporating narrative structure. The folks of the OZ project at Carnegie Mellon University by-passed the issues of interface and computational technology to reach straight for the issues that arise when the "interactor" (their word for the interacting viewer) becomes a character in the narrative (Kelso, Weyhrauch et al. 1993). They built a stage and used improvisational actors to create the story world. Likewise the *Wheel of Life* project at the MIT Media Lab built an interactive set through which participants navigated (Davenport and Friedlander 1994). Both of these projects directly investigated the issues of narrative in immersive environments. But when choosing to use physical sets to immerse their users they gave up the ability to introduce "cuts."

Camera control in virtual environments

Interactive control of the camera in many different contexts has been studied since the advent of computer generated environments. These applications include animation, exploration/manipulation, and presentation. While only some of the most recent work even begins to address the issue of incorporating cuts (and none of the work addresses the issue of cuts in an immersive environment), all of the work provides a background for approaching the camera control aspect of this thesis work.

Early work focused on specifying camera movements over a path. In these systems keyframes are specified by the user and an interpolation method generates a smooth and continuous path for the camera. (Kochanek and Bartels 1984; Shoemake 1985; Bartels, Beatty et al. 1987; Barr, Currin et al. 1992) These methods addressed the needs of animators but did not address the issue of interactive camera control.

The manipulation or exploration applications provided other requirements for camera control. These issues have been partially addressed by a number of different researchers. Some coupled new interface devices with metaphors for controlling the camera (for example the "eyeball in hand" metaphor). (Brooks 1986; Brooks 1988; Ware and Osborne 1990) Others studied new techniques using traditional input devices. (Chen, Mountford et al. 1988) While yet others addressed particular issues in interactive camera control, such as the problem of scaling camera movements appropriately. (Mackinlay, Card et al. 1990) This research will focus on direct manipulation of the camera, not on controlling the camera from a higher level (a task level).

More recent research has focused on controlling the camera based on higher level directives. By coupling camera control with the ability to inquire information about the environment, camera placements can be generated based on visibility and framing constraints. (Karp and Feiner 1990; Drucker, Galyean et al. 1992; Drucker 1994) Of these Steve Drucker's recent thesis work is most pertinent to my research.

Action selection

There has been and continues to be much work in action selection (or planning) in both the areas of AI and robotics. This section is not meant to be a background on this work but rather to highlight some of the work that may be pertinent to this thesis. All of the work surveyed here is focused on reactive planning for agents that manifest themselves as characters in an interactive environment. While characters are an important part of an interactive narrative, my focus is on how these methods can be used to address the action selection issue of plot. If the plot is seen as an agent needing to select an appropriate action to perform next, it is clear how this work would be pertinent.

I have chosen systems that are real-time and reactive, and most of these papers present concrete algorithms. This is not a complete list but serves to point out a few important concepts and lessons. Early work links perception via sensors to create reactive systems, moving away from traditional planning methods (Jappinen 1979; Jappinen 1981; Zeltzer 1983). Maes' work continues this by focusing on a decentralized action selection processes while driving the system with goals (Maes 1989; Maes 1990). She uses a network of links and an energy propagation model which results in the coupling of perception with action. Others expand upon Maes' work in an effort to create autonomous animated characters (Zeltzer and Johnson 1991).

Blumberg builds on the work of Tyrrell (Tyrrell 1993) to incorporate the notion of a lose hierarchy (Blumberg 1994). The hierarchical structure lends itself well to plot. The Oz project at Carnegie Mellon University has also developed a reactive architecture for agents (Bates, Loyall et al. 1991; Loyall and Bates 1993). What is interesting about their work is that it focuses on the use of these agents as characters in dramatic environments.

Geometry and motion representation

The narrative world has to take on shape to be visible to the camera and audience. This thesis will draw on the extensive research in computer graphics on both geometry and animation representation. My survey paper titled *Unified Representations of Geometry and Motion* (unpublished) gives an extensive overview of this material. I will only summarize the material here. In this survey, a unified representation has three parts: the geometry, the method for manipulating that geometry, and the motion model.

Geometry representations divide into two types. These are surface representations which model the surface of the three dimensional object, and volume representations which define a complete three dimensional space or volume. The most common surface models used today are polygons or polygon meshes (Foley, vanDam et al. 1987) and parametric surfaces like spline patches (Farin 1988). It is because of the their ability to be rendered quickly that these surface representations are so commonly used today. Alternatively, volume representations hold more information about the space they inhabit, allowing faster and easier integration into many motion models, but they are often converted to a surface representation before being rendered. There are two basic types of volume representations--analytic (functional) and discrete (sampled). Examples of analytic volume models include "blobby" objects (Blinn 1982), "soft objects" (Wyvill, McPheeters et al. 1986; Wyvill, McPheeters et al. 1986), and super-quadrics (Barr 1981; Pentland, Essa et al. 1990; Pentland, Essa et al. 1991). Discrete volume data is a collection of samples across a volumetric space. Because the discrete volume representation is simple the research often focuses on rendering and manipulation techniques (Lorenson and Cline 1987; Galyean and Hughes 1991).

Geometric manipulation methods are the techniques used to alter the geometry representations. The simplest and most commonly used method is the use of affine transformations (Shoemake 1985; Foley, vanDam et al. 1987; Barr, Currin et al. 1992) that provide a straight forward manipulation of the position and orientation of objects. Other geometric manipulation methods include, boolean set operations (Foley, vanDam et al. 1987; Naylor 1990), free-form deformations (Sederberg and Parry 1986; Coquillart 1990; Hsu, Hughes et al. 1992), and the iso-parametric elements used most commonly in the finite element method (Bathe 1982).

Making these geometric models move is the task of the motion model. Over the years these motion models have evolved from simple position interpolation techniques (Reeves 1981; Shoemake 1985; Barr, Currin et al. 1992), to higher level forward and inverse kinematic methods (Zeltzer 1982; Sims 1984; Zeltzer 1984; Sims 1986; Boulic, Magnenat-Thalmann et al. 1990; Lee, Wei et al. 1990; Badler, Barsky et al. 1991), and now to the physically based simulation of rigid body and flexible body dynamics (Wilhelms and Barsky 1985; Armstrong and Green 1987; Wilhelms 1987; Wilhelms 1987). Most recently, control theory techniques have been used to provide even higher level (task level) control of the motion. These motion models include space time constraints (Brotman and Netravali 1988; Witkin and Kass 1988; Cohen 1992) as well has the forward simulation approaches (McKenna and Zeltzer 1990; Raibert and Hodgins 1991).

Chapter Four

An Analogy for Narrative Navigation

> What is the essence of the director's work? We could define it as sculpting in time.
>
> Andrey Tarkovsky

The narrative space

The presentation of a narrative can be thought of as movement through a narrative space. This is a large, multidimensional space in which all the potential elements and events of a narrative exist. Every potential variant can be indicative of another dimension. The narrative space is a hyperspace. In traditional linear storytelling it is the author's job to navigate this space by steering the audience through the hyperspace in an interesting and provocative way.

The way that the narrative space is navigated is called the plot. Chatman (Chatman 1993) defines plot as the "measured plan or pattern of a narrative." It is important to note that while a plot often moves through physical spaces, it is the axis of time that is most important to how a narrative is revealed. The other element of plot is that of secrecy. Plot takes advantage of the powerful human drive of curiosity. The audiences' attention is held in anticipation of what will happen next. In other words, the audience is looking ahead in the narrative space at the options set forth and is curious as to which path will be taken.

Clearly not all paths through a narrative space are interesting. The author constrains the path that the audience travels to one that is compelling. In traditional linear narratives there is only one path, and it is this same path that is taken every time regardless of how many audiences or how any times an audience experiences the narrative. It is also clear that the same story can be told in a number of different ways. (A great example of this is the 23 different films that tell the story of *The Three Musketeers*.) The problem set forth to the non-linear (interactive) storyteller, is to open up this path. The non-linear narrative has more that one way to traverse the narrative space. Therefore an author of non-linear narrative must construct a representation that constrains the audience to interesting paths while giving them the freedom to deviate from the unchanging line drawn by traditional linear narrative.

Story time & presentation time

> Time is one damn thing after another.
>
> Anonymous

There are two different concepts of time in a narrative. First is the always linear unfolding of the events that the viewer experiences. This is the timeline of the presentation. Then there is the timeline of the story world. This defines the temporal relationship of the story events. These two timelines are seldom the same. In other words, a narrative seldom presents the story as *one damn thing after another.*

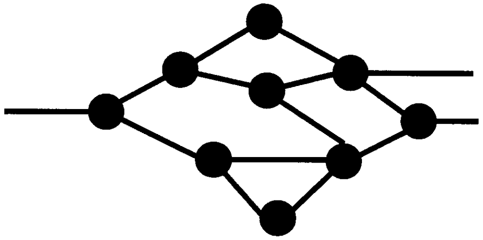
One of the best tools an author has is the ability to manipulate the order in which the story events are presented. This allows the author to build a more affective narrative by controlling what the audience does and does not see, by changing the perceived relationship between the different events, and by controlling what is seen and what is implied.

One of the classic example of this manipulation is the *flashback*. Here the presentation makes a jump in story time (and often a jump in space) to show an event that happened in the past. The position of this flashback on the presentation time-line can be chosen to build a relationship in the viewer mind between the events of the flashback and the events presented just before or just after the flashback. Take an example from *Casablanca*. After Rick encounters Ilsa the film fades into a flashback of the relationship they had in Paris. This flashback serves to explain his unprecedented behavior upon Ilsa's arrival.

Another common example of the separation between presentation time and story time is the use of time ellipse. Take, for example, a sequence where the protagonist enters a stairwell and starts down the stairs. The next shot shows the protagonist exiting the stairwell into the lobby of the building. Here the intermediate action (walking down the stairs) is not shown, but implied. By doing this the filmmaker need not show the audience what is unimportant and the pacing of the presentation can be controlled.

These are two simple examples of the implication of separating presentation time from story time. Many narrative and film theorists have addressed these issues in great detail, providing a much more detailed exploration of the repercussions of this separation (Levaco 1974; Chatman 1978; Branigan 1992). The development of these techniques, pioneered by Sergei Eisenstien (the soviet filmmaker know for his work in montage) and D. W. Griffith (the early American director) (Cook 1990), mark the origins of modern cinema.

Branching analogy



Many non-linear narratives use a branching structure. In this model short non-interactive segments are traversed between nodes. Then a a node interaction dictates which segment is next.

To date many non-linear experiences created have what is commonly called a branching structure. Each branch represents a linear segment traversing part of the narrative space. These segments reach nodes. It is at these nodes where options are provided. The branching analogy is also used in traditional linear narrative (Chatman 1993). A non-linear narrative using a branching structure will travel a branch before reaching a node. At the node the input is processed and the decision is made as to which of the next branches to take. This structure is particularly well suited to digital video. The sequences or shots can be stored on disk. One can be played carrying the audience to the next node, at which point the input dictates the next segment to be shown.

Below are a few of the fundamental problems that often arise from a branching structure. Some interactive narratives have managed to avoid some of these problems while maintaining a branching structure, although these problems are all potential pitfalls of a branching representation.

Discontinuous presentation

Traditional cinematic presentations rely heavily on complete control of time. It is only with a smooth and continuous presentation that a sense of pacing can be created. The pacing can then be manipulated to better affect the audience. Alternatively, it is common practice for interactive branching narratives to stop at the nodes awaiting input.

A segment is presented to the audience and then comes to a halt and waits until the audience provides input. The input is processed and the next segment is shown. This type of presentation chops up the narrative into pieces, which destroys the sense of pacing.

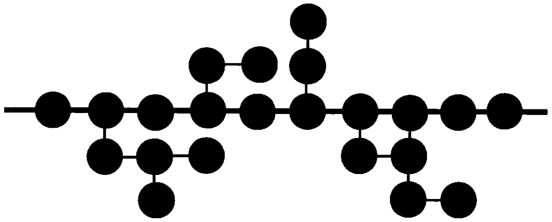
Discrete input

This problem is closely related to the problem of discontinuous presentation. The input in a branching structure is often only available at discrete moments. The presentation either comes to a stop awaiting the input or there is a precise moment at which the input must happen. This makes input events "things" that require the audiences' attention and acknowledgment, segmenting the presentation, and removing the audience from the narrative. By pulling the audience out of the narrative the suspension of disbelief is broken.

This is often aggravated by the fact that input consists of a series of discrete events. The point and click input of a mouse is a good example of this. While there is much information in the pointing process the only information processed as input is the location of the click itself. If input were continuous it could influence and adjust the experience without disrupting the flow or pacing of the presentation.

The "maze"

Take, for example, a CD-ROM based interactive such as *The Journey Man Project* or *Spaceship Warlock*. In these experiences you are left free to explore the world set forth, but there is only one path through the world that will reveal the story. Other avenues are dead-ends. A simple example of this is in the beginning of The Journey Man Project when you are instructed to report to work. If you take any path other than out of your apartment, down the elevator



Some interactive narrative system are structured as a maze. There are many paths but only one leads to a successful completion of the experience.

to the first floor and into the transporter, you come to a dead-end. The system provides a variety of excuses of why you cannot go this way, the most frustrating of which is that you die. None of these side trips are important to the story. In this structure the interaction becomes a puzzle solving problem. You are in a maze and you must find the one true path. In other words, it is your task to find the story.

Two reasons make this a frustrating format. First is the fact that you spend a great deal of time experiencing things that are not important to experiencing the story. (The best films do not show you anything that is not important to the story.) The second is that the character you play is expected to have knowledge that you (as the viewer/interactor) do not have. In The Journey Man Project the player has no idea how to get from the apartment to work although the character clearly does.

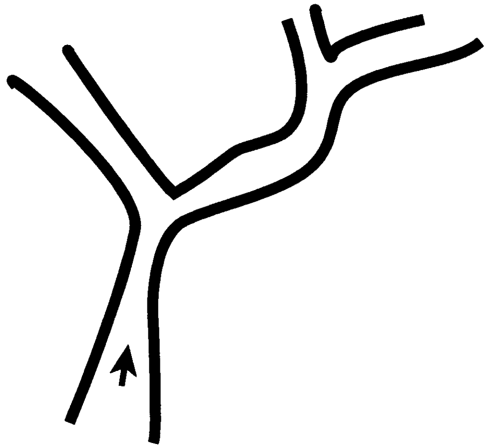
Rather, the audience should only be given choices if the system is ready to make those choices part of the story. In other words, the story finds you regardless of what choice you make.

River analogy

Here I propose an alternative analogy for navigating this narrative space. Instead of linking a sequence of branches and nodes, I suggest that paths through a narrative be more like a river flowing through a landscape. The audience is a boat floating down this river with some latitude and control, while also being pushed and pulled by the pre-defined flow of the water. Like the branching structure, this approach constrains the audiences movement through the narrative to interesting and compelling paths. However, there are some unique advantages to this approach which address the problems listed above.

Continuous flow

The river analogy assures a continuous flow just as in real life. When in a boat you continue to float down the river even when you are not steering. The presentation of the narrative is continuous regardless of whether or not there is input. The amount of control you have over the boat varies with the properties of the river. If the rapids increase you mover faster with less room for swinging from side to side. Alternately, the pace can slow and the river can widen giving more room to steer from one bank to the other.



The river is presented as an alternative analogy for navigating a narrative. The user floats like a boat, able to steer while at the same time pushed or pulled by the flow of the water. This constrains the user to stay within the banks of the river while giving him/her the ability for continuous interaction.

Rudder input

A boat is steered with a rudder. In the river analogy the rudder can be likened to audience input. A rudder takes input continuously. The amount of influence may vary depending on the water conditions but you can always provide that input. Like the river, the structure of a interactive narrative should accommodate input from the audience that can continuously adjust the presentation. However, the rudder does not always have an immediate impact. It may take a while before the input moves the boat from the left to the right bank of the river. Under these conditions input over time has influence over the audience's current position, unlike many branching narratives when a single input event immediately has profound impact on what the audience sees.

No dead-ends

A river cannot have dead-ends. The water must always flow somewhere. Regardless of which fork in the river the audience takes, a viable story is told. Unlike the "maze" problem discussed above the audience will not find itself in a corner needing to backtrack to get on course.

Two levels of structure

The river provides two levels of narrative structure. There is the local structure of the river including the water flow, rocks in the river, the width between the banks, etc., and the global structure including both the path the river flows and the forks that separate and/or rejoin. The audience input has influence on how both levels of this representation are navigated. The rudder or input can steer between the banks, while the position of the boat when a fork is reached will dictate which side of the fork the audience will travel.

Maintaining the plot

So far the assumption has been made that for a narrative to be interactive it must also be non-linear. But is this a true statement? In this section I argue that at some level an interactive narrative should be linear. Plot is the action of moving forward. It is this movement forward that takes the audience through moments of change, i.e. the plot points. There is no reason to believe that the plot need be scrambled to allow for interactivity. There is a great deal of freedom in the process of narration that can be exploited without it being necessary to make the plot non-linear.

Narratives have many different levels of representation. This idea of different levels of narrative representation was introduced earlier and will be an important part of the next chapter. There is the low level representation that details the elements of the presentation. But there are also high level representations that we experience only indirectly via the presentation. This high level is called plot. At its highest level plot holds only the key ideas of the story. Think about

Raiders of the Lost Ark for a moment. This film might be described as *good vs. evil*. At a slightly lower level a representation of this film might consist of the following five parts: the audience learns what type of person Indy is, Indy gets a lead on the whereabouts of the long lost Ark, he finds the Ark, he fights for the Ark, the Ark is lost again. The core structure of the narrative is represented in this form, but there are a number of different ways that the actual presentation can play out.

This notion of higher levels of narrative representation is not a new one. The narrative theorist Todorov argues that in its most basic form a narrative consists of five stages (Todorov 1971). Branigan, working from Todorov's and others' work, introduced a better understanding of these high levels of representation by developing a narrative schema (Branigan 1992). Many directing and screenplay writing books push the idea of taking a single *central theme* and unfolding it into a sequence of acts (most often three acts) (Field 1984; Richards 1992; Hunter 1993).

But is there a high level representation of interactive narrative? Does it need to be flexible and non-linear? We know that the lower level of representations (those that include the presentation) must be flexible and change in reaction to the viewers' input. An example might be where you are a character in the narrative world. You are sitting in your apartment when you get a call. The person on the other end explains to you that you are going to have to come back to the agency for this one last job, whether you like it or not, and to be in the lobby in five minutes. The high level representation of the next part of the narrative is that you will end up at the agency. We can think of this not just as an explanation of what will happen but also as a goal set forth by the plot. And this goal could be reached in a number of different ways. As the interactive viewer, you have a number of different options. You can go directly to the lobby and be taken to the agency. You might try and slip out the back door at which point someone will be there to pick you up. There may be many different avenues you could take but the net result is always that you end up at the agency. The plot goals is always satisfied. In other words, the *what* is predetermined, while the *how* is determined by the interaction. This approach models the narrative as a high level linear set of goals that gives the freedom to the user to influence how these goals are reached. The Oz project at Carnegie Mellon University has taken some step toward this goal with their plot graphs (Kelso, Weyhrauch et al. 1993), but they have not built a computational model.

Under these conditions the interactive narrative finds you regardless of where you go. The narrative shapes and guides your experience through the narrative space, pulling you back on course. It is, in effect, "sculpting in time." In this model of narrative guidance of interactivity, two different audiences should have a common experience at a high level while the details of the experience may very widely. They might expound about the irony of the story or how much they enjoyed the dog, while they would be surprised to learn that they do not share details of the presentation (like whether the dog jumped on them).

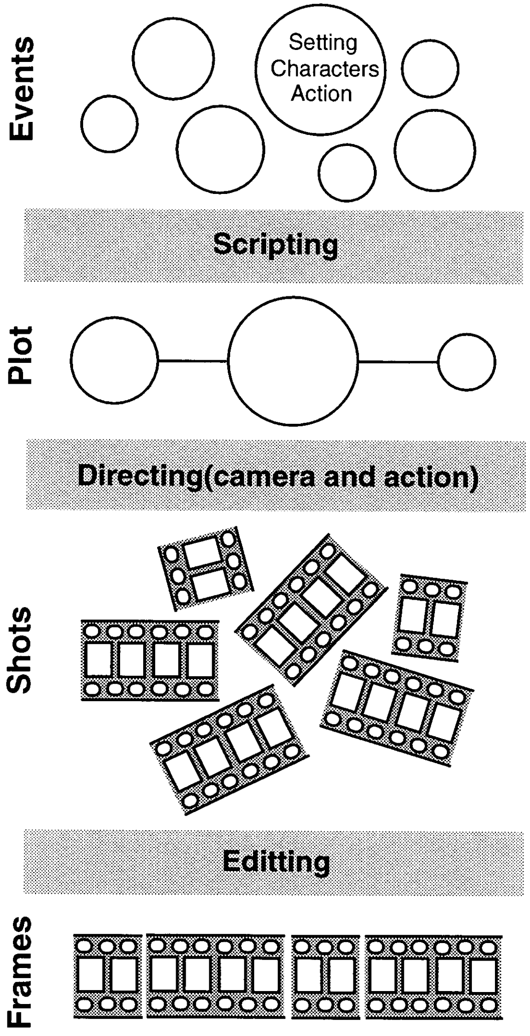
Chapter Five

Representing Interactive Cinema

Representations of a film

Just like any narrative, the narrative in a film has many different representations. As the audience we are used to seeing the one represented as frames, that final strip of film consisting of frame after frame of images to be projected on a screen in a dark room. But there are other representations of a film's narrative. There is the representation that we, as viewers, build and store in our heads, that allows us to recount the narrative for ourselves or others. There are also a number of representations that the narrative goes through during the production process. These include: the treatment, the script, story boards, dailys, the shots, and the sequences.

The diagram on the next page is a stack of different narrative representations for a film. It is important to note that although the diagram includes narrative representations often used in the



production process, the diagram is not representative of the production process. This diagram includes (starting at the top), events, plot, shots, and last frames. Between each of these representations a gray box is indicative of a method by which the representation above a gray box is converted into the representation below it.

There are two things to note about this diagram. The first is that with each higher level representation, more information from the final narrative or film is missing. For example the plot level representation contains little information about the framing or camera motion. The information that is missing must be reconstructed before the narrative can be presented to an audience.

This second observation is the fundamental difference between the top two representation and the bottom two. There is a distinct line in the middle of this diagram. Above the line are the event and plot level representations containing the core element of a narrative. Below the line the representations of shots and frames embody the details of discourse/presentation. The event and plot level representations are based on the fundamental unit of an event. The lower level representations use individual images or frames as their fundamental unit. Both the final footage and pool of shots from which it is assembled are the product of this need for frames in the final product.

Events

A film consists of a chain of events that is presented to the audience. These events can be separated into elements independent of their relationship to each other. Bruner (Bruner 1986) emphasizes the need for each event to be significant to the narrative with his concept of "vicissitudes of intention." Like many script writing guides instruct (Field 1984) Bruner states that every scene, every fragment, must further the development of the narrative. Each of these events is representative of a change in intention or a "vicissitude of intention." An individual event has associated with it a setting, character(s), and actions to be performed or to happen. In Chatman's narrative structure *story* is separate from *discourse*. Story is only the actions, happening, setting and characters of a narrative. Therefore these events collectively represent Chatman's *story*; the narrative without the discourse. In other words, these events are a pool of elements that are drawn upon and laced together to build a narrative.

Plot

The scripting process chains together the events. This chain or sequence is called the plot. We know from experience that events do not have to be presented to the audience in the order in which they happen in the story. Flashbacks are a clear example. So there are two separate time-lines, the story time-line and the presentation time-line. The plot dictates each event's location on the presentation time-line.

Shots

A director uses the camera to give the audience a window into the events that take place in the plot. The director is making detailed decisions about the discourse of a narrative by deciding how the plot will be shown to the audience. This includes both the details of the cinematography (such as framing and camera motion) and the details of the action (for example whether or not an actor will use his left or right hand.) The result is a collection of shots. These shots are a representation of the film. A representation that does not contain information about presentation ordering. Some of the ordering is detailed by the plot level representation, but there are still decisions to be made about sequencing.

Frames

The editing process builds a linear presentation from these shots. This is a process of selecting the shots to use and assembling them. The result is a sequence of frames.

Non-linear video

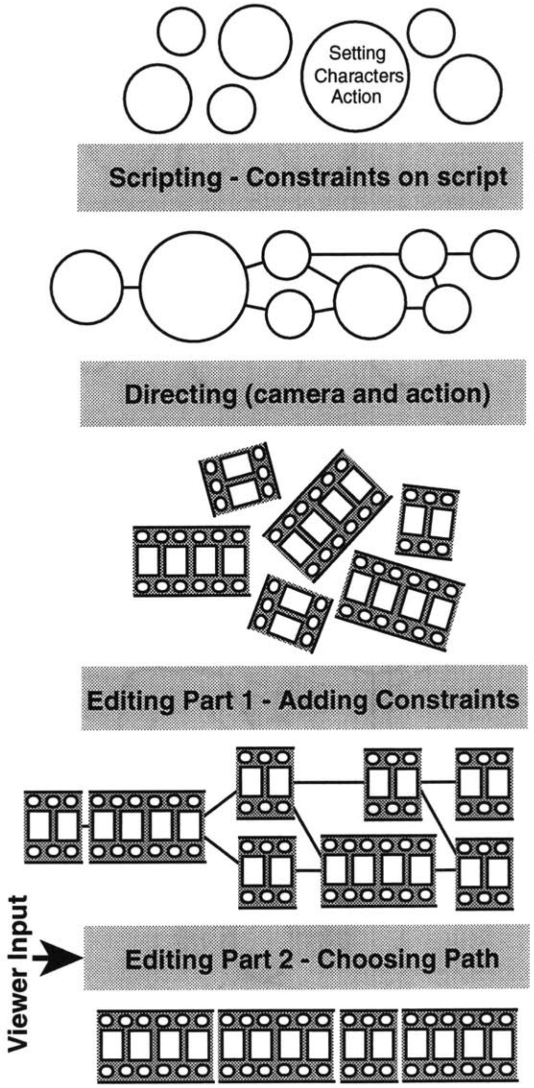
A large number of interactive narratives that have been produced to date (and continue to be produced today) use some sort of non-linear video footage as their core technology. Digital video, for example, can be stored on disk and randomly accessed to bring up any stored shot or frame to the screen at any moment. In other words, all the tools traditionally made available to the editor of a linear film become part of the tools used to make a narrative interactive. There are a number of examples of this type of interactive experience. The *Aspen* project from MIT's Architecture and Machines Group (Mohl 1981) and commercial products like *SpaceShip Warlock* and *The Journey Man Project* are interactive experiences that allow the audience to navigate a virtual space with a first person relationship to the material. Alternatively, projects like *New Orleans* (Davenport 1981), *An Endless Conversation* (Davenport, Evens et al. 1993), and *Trains Of Thought* (Halliday 1993), are examples of navigating fictional or documentary databases of footage as an outside observer.

In these applications a plot is constructed with options. The plot provides more than one way to take the audience from the beginning to the ending. The more common structure for these flexible plots is a branching structure (the illustration of plot in the diagram is representative of a branching structure.) Recent research has introduced new methods for constraining the plot from the potentially infinite possibilities, in particular the use of filters (Evans

1993).

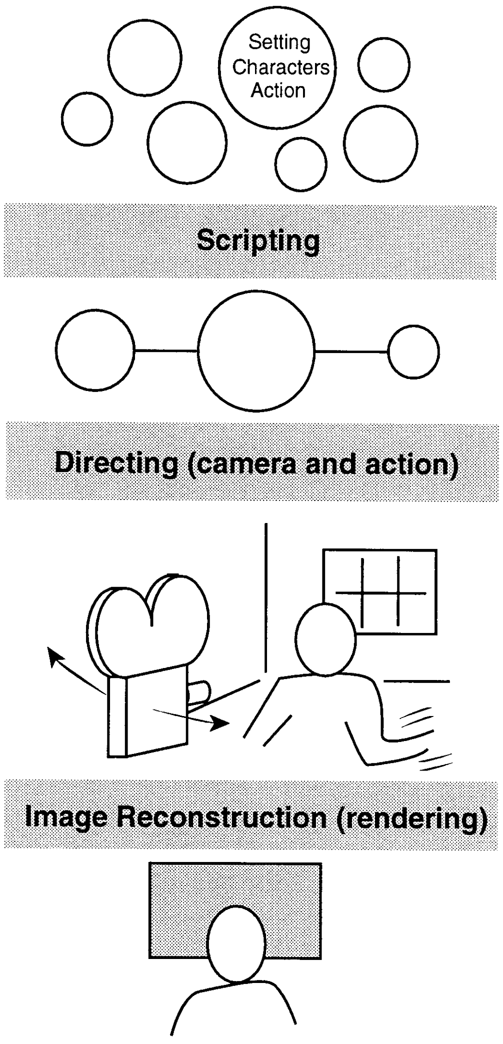
Once a flexible plot structure has been created the next step in a non-linear video is to generate shots for all the possible variations in plot and presentation. Of course just like in some of the film representations discussed above, the shots are dictated by their basic unit of a frame. These frames are then stored for later access. With the shots created they are linked together for play. Often the same structure apparent in the plot is transferred to a representation that links the shots together. Now viewer input can complete the process. Viewer input dictates which linear presentation happens.

Even with today's technology the storage of frames is extremely expensive. For this reason, interactive video narratives are often restrictive just because of storage limitations. But even if storage was limitless the presentation would be limited by what frames are stored. The audience can only be shown the frames/shots that are produced and stored in the database. If the interface wants to accommodate a plethora of subtle changes in presentation then all the necessary shots and frames must be staged, directed, shot, cataloged, etc. This can be prohibitively expensive.



No more frames

In both of the previous sections many of the representations presented and discussed are a product of the need to generate frames. Both video and film rely on the product being stored as a sequence of images or frames. Frames have been such an import part of these mediums for so long that they have dictated / influenced many of the narrative representation used today, as can be seen in the previous two diagrams.



Because frames are so pervasive in traditional linear video and film it was only natural that frames be used in emerging interactive systems. The decision for frames to be used in these interactive systems was also governed by the fact that so much of the technology (both display and storage) was built around the concept of frames.

It is important to realize however that the need for frames is strictly a product of the display technology. The mechanics of both video and film display are possible only because a series of frames presented at a quick enough rate are perceived by us as a continuously moving image. A movie projector flashes 24 different images on the screen a second while a video display scans the surface of the CRT 30 times a second. While we would not be where we are today without frames, there are clearly some limitations to their use.

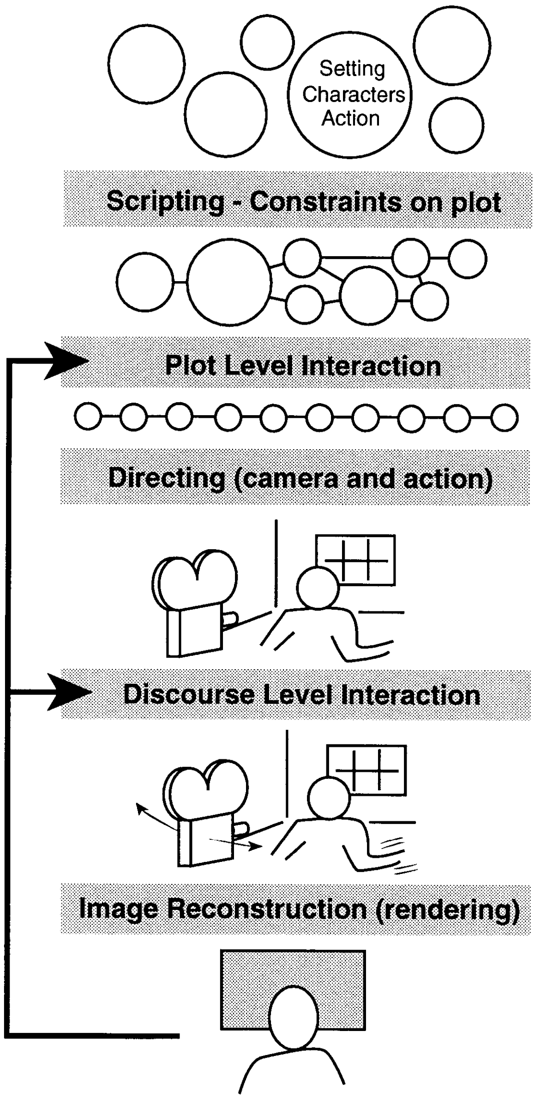
As all our media becomes digital, frames will be expensive to store and transmit and are not easily searched when one is looking for something in particular. But even more important they are not flexible. A narrative stored as frames is not designed to be changed and altered. While the frames can be re-ordered (edited) to accommodate change, the individual frames are fixed and can not be altered to accommodate viewer input or preference. In short, frames are not amenable to content based representations of narrative.

Fortunately today's technology can provide us with some alternatives to the use of frames.The basic idea is to represent the geometry of the objects in the scenes and these objects' motion directly. This includes representing the camera's geometry and motion. The geometry and motion can be stored more compactly, with content based information, and with infinitely more flexibility. This type of representation is also display method independent. From a geometry and motion representation, frames can be generated for viewing on conventional displays. In addition, alternative display technology can be accommodated i.e., head mounted and holographic displays. These new display technologies inherently move away from the notion of static, pre-recorded frames and toward a more immersive environment where the camera becomes the viewer's eyes.

This idea of a model based representation is not a new one. The computer graphics community has long researched the problems behind geometry, motion representation, and synthesis, as well as the issues of image generation (rendering) from these models. Moreover, computer vision researchers have long focused on the problem of extracting these models from existing video sources. More recently, researchers have begun to marry these two worlds in order to create what is often called *structured video*. (Bove 1994). The process of structuring video is the process of generating these models from a traditional linear piece of video frames. These structures, while having all the advantages listed above, can be used to reconstruct (or render) the frames needed later for display. In a structured video model, frames are no longer part of the representation and are only generated as part of the display or presentation process.

Two levels of interaction

The next question is what kind of impact does this move to model based representations have on the interactive narrative? The display process can now be thought of as instantly generating any frame that is needed and displaying it for the audience. In this way a whole new set of variables can be changed directly or indirectly by the viewer during the presentation process. There are many attributes these new variable can govern; for example, the camera parameters, including things like where the camera is positioned,



what is in the frame, where in the frame it is and how the camera moves.

At this point let us take a look at a new diagram of representations. Like the earlier diagrams this one shows a stack of different representations and the gray areas between them are the methods by which one representation is transformed to the representation below it. As before the highest level is nothing more than a pool of events, from which the narrative is constructed. And like before a set of constraints is imposed on these events restricting the ways in which the viewer can navigate the narrative.

Plot level interaction

But now we recognize that viewer input can influence the narrative at this plot level. The viewer directly or indirectly dictates which of the possible chain of events is to be traversed. The result is a linear chain of events to be played out. For example this input could dictate the order in which the narrative events are presented or which events will be used to convey the necessary information. Will the key event of the protagonist's childhood be presented at the beginning or will it be held to later and presented as a flash back? Will an event be shown directly to the viewer or will it be implied? Will the viewer see Joe hit by the bus or will they learn of Joe's accident from a news broadcast?

Presentation level interaction

With a chain of events decided, these events must be performed so that the audience can see and hear them. It is at this stage that the plot elements are coupled to the geometry and motion models. Just like during the production of a film there are many decisions about how these actions will be presented. There is still a lot left unsaid at the plot level of representation. Conventionally directors and editors make these decisions during the production process. Instead, we have the freedom to let these details change and adjust with viewer input. So like the constraints placed on the plot of an interactive narrative, constraints can be placed on these presentation details. This leaves room for a second level of interaction whereby the viewer influences the details of the presentation. Details of the camera, character movement, or props can become free to change. Will the next shot be a close-up or a medium shot? Will the character reach for the gun with her left or right hand? Is his hat red or blue?

In this new structure there are two levels at which a narrative can be interacted with: the plot level and the presentation level. At the higher level a viewer steers his/her way through the plot. It is at this level that the viewer's relationship to the narrative is exercised. At the lower level a viewer alters the details of the look and feel. It is at this level that the viewer's relationship to the presentation is dictated.

A representation for narrative guidance of interactivity

The last four sections have introduced and discussed different levels of representation of cinematic narrative. In some cases the discussion has also explored how these representations are related to and influence each other. This discussion started with a structure for traditional linear narratives like movies. It was then shown that these representations, like those used in non-linear video narratives, are greatly constrained by their reliance on frames. Then the more flexible representation of geometry and motion was introduced. This brought the discussion to the realization that there are two levels at which input can influence a narrative: the plot level and the presentation level. But none of this says anything about how such an interactive cinema can be realized. To do so there is a need for a framework. A framework which defines the elements needed to build an interactive narrative. This piece of the puzzle must be defined and a structure must be created to illustrate how these pieces can fit together. Such a framework was presented in the approach chapter. Here I want to revisit that framework to illustrate how it grew out of the representation discussion in this chapter.

Joe Bates' discussion in *Virtual Reality, Art, and Entertainment* (Bates 1992) lends some insight into what the pieces of this puzzle might be. He begins by looking at what has made the fantasy worlds of novels, television, and film so successful. He identifies three key elements that are present.

- Living creatures, usually human, and usually embodying some intelligence and emotion.
- Long term structure to the events portrayed, in other words a story is told.
- An effective and emotional presentation.

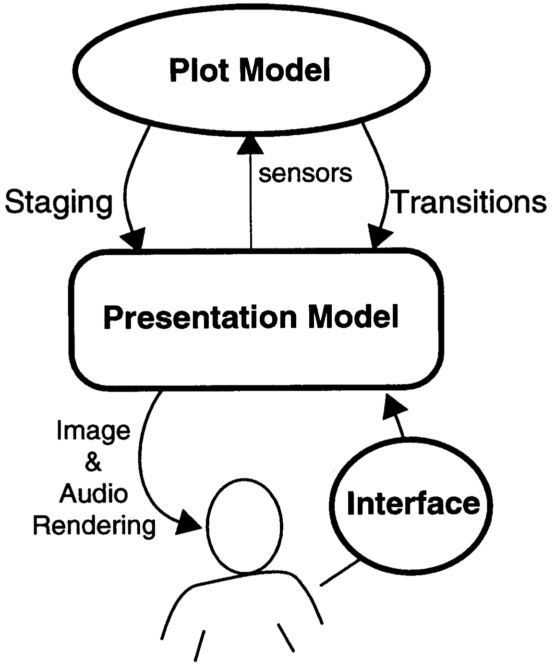
Bates goes on to identify three areas of research that directly correspond to these elements listed above: construction of computational theories of agents (characters), presentation, and drama. A complete computational model of a character includes the geometric models, the motion models, and the action selection (behavioral) models of the characters. This discussion serves to define three important areas of research. While closely related, I propose a slightly different breakdown of the interactive cinema.

I define three components needed in an interactive narrative. The first piece is the plot model i.e. the highest level representations. Second, is the pieces of the presentation level. At this level the characters, camera, sound, music, etc. are represented. Last is the interface through which the viewer transmits information back into the narrative. The diagram illustrates the relationship between these parts. The plot model rests on top, shuffling information into the element of the presentation level. Together these elements can be used to construct images and sound for the audience. This comprises the flow of information from the narrative to the audience. It is through the interface that information flows from the viewer back into both levels of narrative representation, plot, and presentation.

This framework as a whole is a model of an interactive narrative. This large model is separated in to the pieces listed. Each of these "pieces" is a model in and of itself representing its "piece" of the interactive narrative framework. In these components the term "model" refers to both the data and computation process which takes information in and outputs information to another model or directly to the audience.

Plot model

Just like the plot representations discussed in previous sections the purpose of a plot model is to provide a method by which narrative events are connected to each other. This plot model provides a structure from which a presentation order of events can be extracted. In other words, the model dictates which events and in what way the events, from a pool of potential events, can be chained together, eventually producing a linear narrative that will by viewed be the audience. In a traditional non-interactive linear presentation this model is nothing more than a linear chain of events. The chain specifies each event's location on the presentation time-line. Alternatively, a plot model for interactive narrative is a structure that



This diagram shows the basic structure of a narrative guidance system. At the top is a plot model. This is then coupled to the presentation model via sensors and direction (staging and transitions). The presentation level contains models of all the visual and audio elements. The interface provides the link between the user and the presentation level.

provides different ways in which the events can be chained together. Under most circumstances the structure of a plot model imposes constraints on how these events can be combined; constraints like event A must come at some point before event B, event A or event B but not both, amongst others.

The narrative structure or plot can be thought of as an event selection process as well. There is a pool of events that are potentially part of the narrative. The process is one of selecting which events and in what order they are to be presented. In a film this task is done up front by the director. The presentation style greatly influences the mood and/or feel of the experience. In film this includes the lighting, the camera position, the framing, editing, etc. In an interactive narrative this presentation is greatly influenced by the interface. For example, if a head-mounted display with a head tracking device is employed, there is a kinesthetic connection between the viewer's head orientation and camera's orientation.

Presentation model

The presentation level includes the elements needed to construct the presentation including the visual, audio, tactile, etc., components of the presentation. The emphasis in this work is on the visual presentation, such as the geometry of the sets, props, and characters, their motion, and the geometry and motion of the camera through which the narrative world is seen. In traditional film and video the presentation model is completely contained in the "frames." The frames contain all the necessary geometry, motion and camera information needed to produce a presentation of the narrative. As groups the frames produce larger elements, shots, and sequences. Each frame records and stores the geometry, and a piece of the motion of that geometry from the vantage of the camera. In other words it is images (frames) that are the representation format of the geometry, motion, and camera.

While a frame based model is visually rich it lacks flexibility. The earlier discussion illustrates that a narrative stored as frames cannot economically allow subtle adjustments in geometry, motion, or the camera. Because of these limitations, the framework presented here assumes an alterative presentation model, like the frame-less representations discussed earlier. When using content based models for representing a narrative it is advantageous to separate the geometry and motion models from the camera model. This framework does just that.

Note that in this new framework the display process has changed. In traditional film the display process is nothing more that the projection of images on a screen. Here those images need to be created and then displayed. The process of rendering takes the geometry and camera information at a given moment and generates the image (frame) needed for the audience to see the narrative. What is important about this is that this rendering of the images is now a part of the display process not a part of the narrative model.

The interface

The plot and presentation models are, as a whole, responsible for transmitting information from the narrative to the viewer. Information flowing this one direction is only half of what is needed to make a narrative interactive. For anything to be interactive, information must flow in both directions, and the interface is the means by which information flows from the audience back to the narrative.

The interface is both the hardware and the software that ties the physical actions of the viewer (his/her input) back into the narrative. Take, for example, a single button. The action of the viewer pushing the button is converted by the hardware to a signal that is then received by the software in the computer. Having received the signal the software can then interpret the signal and influence the narrative based on the current state of the narrative. In this example the interface is comprised of the button, the circuitry generating the signal, and the software interpreting the viewer's action. All the pieces of the interface translates the viewer's actions into information that guides the narrative models toward a particular experience.

In the framework given here the interface is shown influencing only the presentation level. Previous sections talk about the interface directly affecting the plot. When the user is directly influencing the plot the user is changing the story. This is appropriate for some models of interactive narrative but is not consistent with the narrative guidance approach. To assure a given story is told the plot must be free from direct user input. Therefore this approach restricts the user's influence to the presentation level. This input then indirectly reaches the plot. The plot continually monitors the state of the presentation with sensors and uses that information to adjust the presentation via staging and transitions.

Chapter Six

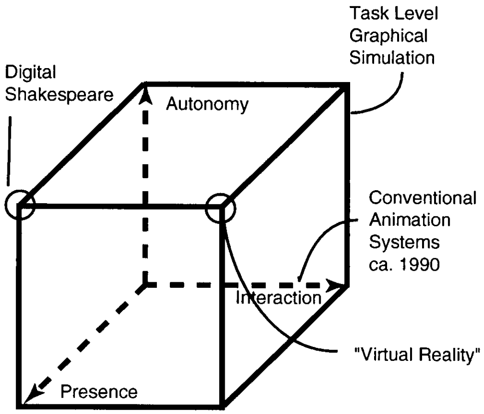
A Taxonomy

> Solving a problem simply means representing it so as to make the solution transparent.
>
> Herbert Simon

A taxonomy classifies a set of organisms or objects. In this taxonomy "interactive experiences" and more specifically "interactive narratives" are being classified. But there are many ways that one set of things can be classified or organized. To what end then, are these interactive experiences being organized? First and foremost is the reason stated so eloquently above. That is to make the problem easy. To reveal the solution. It can be argued that finding the correct perspective on the problem is more than half the battle to finding the solution.

The second goal of this taxonomy is to provide a context in which this thesis work can be cast. In this role the taxonomy is a map of existing work and it shows the area in which the new work will focus. It will also reveal the relationship of this new work to the existing work. But as Laurel notes (Laurel 1989), it is important not just to extrapolate from current interactive entertainment. Current work has limitations and if all that is done is to extrapolate from existing work, it is likely that the limitation of that work will be carried on to the next stage.

There are two taxonomies that are pertinent to this work. One by David Zeltzer (Zeltzer 1992), *Autonomy, Interaction, and Presence* and one by Brenda Laurel (Laurel 1989), *A Taxonomy of Interactive Movies.* Zeltzer's AIP Cube is a classification of graphical simulation systems. In this taxonomy there are three criteria: autonomy; the amount of computational models simulating a behavior, interaction; the means by which the simulation is modified, and presence; the channel over which the simulation is conveyed to the participant. These three variables define a space (a cube) for which he discusses what type of simulation exists at the corners. As can be seen in the diagram, the "grail" resides at the location with maximum value for all three criteria. An experience that resides at this rich location, in the AIP cube, is referred to as a "virtual reality."



Above is Zeltzer's AIP Cube, a classification of graphical simulation systems. The three axes represent autonomy, presence, and interaction. While arguably these properties are also properties of interactive narrative this taxonomy does not address the issue of narrative structure or plot.

While these elements of a graphical simulation system are closely related to the elements of an interactive narrative, a graphical simulation system does not necessarily provide any narrative structure or plot. It is this dramatic structure or relationship between content and structure that provides the success of a good narrative. Therefore a taxonomy about interactive narrative will have to say something about narrative structure. Zeltzer touches on narrative structure with one example, "digital Shakespeare." "Digital Shakespeare," implies the existence of a narrative structure. His description is a good example of a graphical simulation system with high autonomy and high presence. But it does not speak to the fact that narrative structure is an essential or even important part of a graphical simulation system. With that in mind it is easy to see that a graphical simulation system could be classified at any location in the AIP Cube (including the "virtual reality" location) regardless of whether or not it contains any narrative structure.

As the title of her paper (*A Taxonomy of Interactive Movies*) suggests Laurel does incorporate the need for narrative structure. She argues that plays and movies have structure and that it is the relationship between this structure and the content that makes them successful. Laurel's taxonomy uses three definitional variables: activity, interactivity, and personess. Activity, which is an expression of how the user relates to the interactive movie, is the variable which expresses the narrative structure the most. The activity variable can have one of three discrete values: explore, control, and enact. These values suggest different levels of narrative structure. For example the experience of exploring would not be expected to have much narrative structure, while the process of enacting would suggest that you are a character in a larger structured experience. These terms also suggest that the viewer has a particular relationship to the experience. Both the words enact and explore suggest that the viewer is present in the narrative world.

The problem with this formulation is that there is no distinction made between the viewer's relationship to the characters, the interface's influence on the different narrative elements, and the camera's relationship to the characters. I suggest that these are three different issues. But while each of these issues may have an influence on the others, it is still beneficial to treat them as separate variables in the taxonomy. By separating these variables, a larger and richer set of interactive narratives can be represented in the taxonomy.

There are a number of video games where you, as the player, are one of the characters in the games. The driving games are a clear example of this. You are sitting in the driver's seat of a car looking out of the eyes of a driver. But it is not always necessary for the camera's relationship to the characters to be coupled with viewer's relationship to the character. The player does not have to look through the eyes of the character in order to have this one to one relationship with the character. An example could be one of the many video games where the player is Mario, Sonic the HedgeHog, or one of the many fighting characters so popular today. In all these examples the player is one of these characters. Their every push on the game pad has a direct influence on what their character is doing. But in each of these examples the camera is an on looker showing the action. In other words, the variable that governs the viewer's relationship to a character is independent of the variable that governs the camera's relationship to the character.

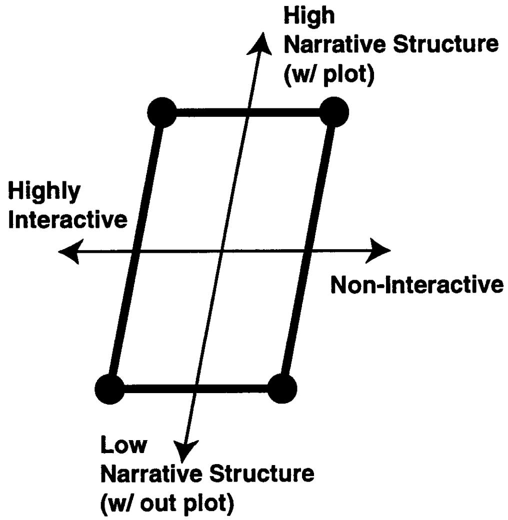
Interactivity	**(high -- low)**
Narrative structure	**(high -- low)**
Interfaces Influence	
Plot	**(direct/indirect/none)**
Presentation	**(direct/indirect/none)**
Camera	**(POV/on looker)**
Viewer	**(character/god)**

This taxonomy considers the above variables with respect to interactive narrative. First is the spectrum along which the amount of interactivity and narrative structure varies. Second is the issue of user influence on both plot and presentation. Last is the camera's and the user's relationship to the story.

With these issues in mind the taxonomy proposed is presented in the next three sections. The first section develops a map of interactive narrative based on the two key variables of interactivity and narrative structure. The second section addresses the issue of how the interface relates to the many different elements of a narrative at both the plot and presentation level. The third section clarifies the separation of viewer's relationship to the characters and the camera's relationship to the narrative elements as being issues not just separate from each other, but also separate from the issues discussed in the first two sections.

Finding interactive narrative

The two primary distinguishing factors of an interactive narrative are the amount of interactivity the viewer has with the experience, and the amount of narrative structure the experience imposes. The diagram looks at these two variables. The two axes map a space of interactivity and narrative. This map ranges from the extreme of an unstructured exploration of daily life to the highly structured presentation of a suspenseful film; and from the highly interactive routine of our daily lives, to the passive activity of watching television or a film. Before exploring this map and charting the location of different interactive narratives, we must first explore what narrative structure is and what are the aspects to making something interactive.



Above is a visual representation of the two axis of interactivity and narrative structure. This taxonomy presents these as independent variables allowing for experiences that are both highly interactive and have narrative structure.

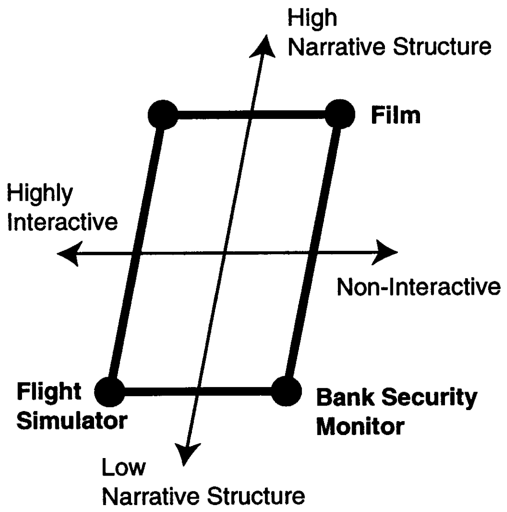
What does it mean for an interactive experience to have narrative structure? What is narrative structure? A number of film and literary theorist make the distinction between the "material and the structure" (Branigan 1992), the "story and the discourse" (Chatman 1978), or the "stuff and vicissitudes of intention" (Bruner 1986). These are all ways of separating the physical elements that we see and hear during the presentation of a narrative from the structure that is revealed to us over time. Narratives are unfolded for us over time. We watch and listen and piece together these elements to build a complete drama (story) over the course of the presentation. The way these bits of information are presented to us is the structure of the narrative. This structure or plot grows over time; it is a time based manipulation of the presentation. It is only because of this temporal quality of the plot that such dramatic phenomena as *expectation* and *suspense* can be created. Narrative structure is the temporal relationship of the events presented to the viewer which gives intensity and meaning to the narrative world.

Traditional narrative forms like film and books transmit information from the narrative to the audience, but do not often receive information from the audience. It is only once this second channel, allowing information to flow back into the narrative from the audience, is established that a narrative becomes an interactive narrative. The levels of interactivity are a continuum. It might be that the viewer can only input a very small amount of information occasionally and this information has little influence on the narrative itself. Alternatively, a viewer may continuously have an infinite number of ways to profoundly affect the narrative. This situation shows that there are a number of properties that define interactivity. Andy Lippman gives a concise and eloquent definition of five different elements of interactivity. (Anderson 1989) Lippman bases his discussion on the give and take of a conversation, arguing that the following five properties must be part of a conversationally based interactive experience.

- **Interpretability.** Each individual in the conversation has to be able to interrupt the other.
- **Granularity.** The size of the smallest element from which the interaction is built.
- **Limited look-ahead.** There must be a limited reliance on any ability to precompute, because the nature of interactivity and conversation is to change and adjust constantly.
- **Graceful degradation.** Requests that cannot be addressed should be gracefully deferred.
- **The appearance of infinitude.** The system should provided the illusion that there are an infinite number of alteratives.

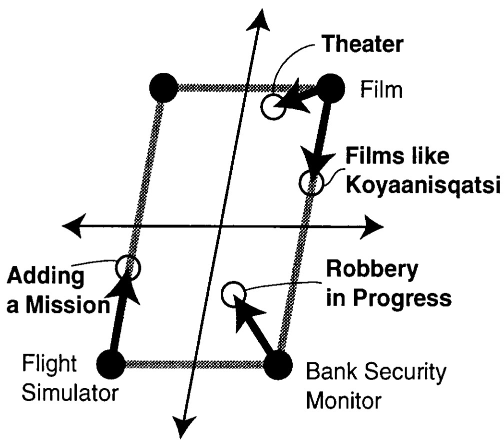
The better an interactive narrative achieves these five goals, the richer the interaction with the narrative. In this map of interactive narrative, a system is considered to be more highly interactive if it more fully meets these five criteria.

We can begin to populate the space of interactive narrative in the diagram by first looking at the corners of the map. The right hand edge of the map is the line along which narrative with no interactivity lies. The upper right hand corner (the top of this line of no interactivity) is the location of high narrative structure. Many films are located in this spot. Moving down the right side narrative structure is removed. The bottom right corner is the location for experiences that have little or no plot and no interactivity. The process of passively observing a bank security monitor would be a good example of this. There is no structure guiding the actions of the people in front of the camera and the viewer has no control over what they do or how s/he sees the action. Moving left, interactivity is added. The lower left corner still contains little or no plot but it is highly interactive. Experiences in this part of the map are not unlike the daily routine of our lives. Flight simulators are a good example of an interactive system that resides at this corner. A flight simulator provides a free and open environment where the participant is free to explore. This leaves the upper left corner. The location on the map where a system will have both high interactivity and a large amount of narrative structure. To my knowledge, nothing to date has the properties needed to be mapped to this location. It is in this area that I and others are pushing toward when we strive to create a truly interactive narrative.



At the corners lie: film — non-interactive and highly structured, video security — neither interactive or structured, and entertainment based flight simulators — giving the user a great deal of freedom but providing little or no story structure.

With the corners established let us look at some examples of systems that push away from the corners. This will help define the contents of the map, and begin to more fully populate it. A film that resides in the upper right corner is a film like *The Silence of the Lambs* which has a very highly structured set of events. Films like this rely heavily on the order and method of their presentation and performance to achieve their high dramatic impact while remaining intelligible to the audience. Alternatively, films like *Baraka* and *Koyaanisqatsi* take the viewer on a journey by presenting sequences of images and sounds. By their nature the success of these films does not rely on strong control of their narrative structure. Films like this are still positioned along the right edge but lower down on the right hand side because they have less narrative structure. Theater (stage production) is, more often than not, highly structured like the first films discussed. But there is something unique about theater that makes it interactive. The best stage actors watch the audience and adjust the subtleties of their performance to better influence the audience. In this way the audience is unconsciously providing input to the narrative. This little bit of interactivity moves theater off the right side of the map.



Flight simulators become more structured when a mission is added, giving the user a goal. Theater is more interactive than film because it allows the presentation to vary with respect to audience response. The experience of watching a security monitor becomes both more structured and more interactive when a robbery is in progress. (The user then has a decision to make as to how to react to the events.)

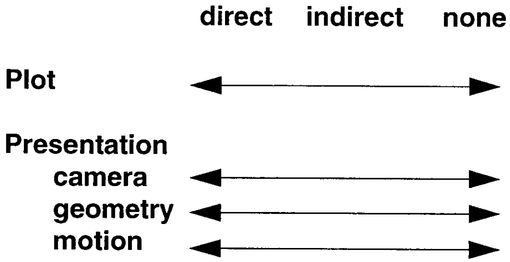
The process of watching a bank security monitor during more daily routines is neither interactive nor does it present any narrative structure. But how does this viewing experience differ if there is a robbery in progress? In this case the viewer is looking in on (through the camera) the activities of a highly structured plan. The goals and intentions that the robbers have served to structure their actions as the narrative is played out in front of the audience or security guard. In the role of the security guard the audience has the ability to interact. The guard can take actions to try to stop or foil the robbery. Under these circumstances the experience of watching the bank security monitor is both more interactive and has more plot, moving its location toward the center of the interactive narrative map.

Early games based flight simulators are classic examples of a highly interactive experience that is free exploration of a synthetic environment. In the simple flight simulator world the user can take off, fly around, land, and/or crash. The only narrative structure in this environment is that imposed by the user. The user may have a set of goals or a plan for their experience that can add structure to the experience, but the system does not provide any structure. There is no narrative in which the user has his/her experience. The only narrative is one that the user makes up as s/he interacts. When someone uses a commercial flight simulator they are often given a mission a goal for the flight. There are also a number of recent flight simulator games on the market today that impose a mission on the user. By adding a mission to the experience, a set of events unfold over time to guide and direct the action of the user. This event can be orchestrated to lead and guide the user through a suspenseful plot. The mission adds narrative structure, moving the system up on the map.

With the exception of the arrow pointing out films like *Koyaanisqatsi,* the arrows shown on the map move toward making narrative interactive. They all point toward the upper left corner, the location on e map where narrative structure (plot) and interactivity succeed and co-exist. This map does not show us what kind of system exists in the golden area of the upper left, but it does show the surrounding area and the different angles from which that area can be approached.

The nature of the interface

Regardless of the level of interactivity and narrative structure in a system, there is still the interface and how it is connected to and affects the narrative. The interface's influence on the narrative specifies how the viewer can affect the different elements of the narrative, including the plot, the geometry and motion (action and performance), and the camera. Can a viewer prevent a car crash? Can the viewer turn the camera to see another part of the room? Can the viewer affect whether a character picks the gun up with his right or left hand?



A user's input can be linked to the plot and/or the many different parts of the presentation. The type of link will dictate if the interaction is direct or indirect.

The previous chapter separates narrative into two parts: the structural part; the plot, and the pieces that present that structure. Likewise this section will separate the discussion of the relationship between narrative and interface into two parts. First is how the interface affects the plot. Second, how the interface influences the presentation elements, in particular, the geometry, the motion, and the camera. There are three levels at which an interface can influence one of these aspects of a narrative: **directly**; the user's actions have a direct and often immediate affect, **indirectly**; the user influences the narrative in a roundabout way, and **none;** the user's input has no influence on that part of the narrative. It is important to note that difference aspects of the narrative can have different relationships to the interface. For example the user may no or little indirect influence on the plot while having complete control over the camera. This ability to separate the different parts of the narrative interface provides a rich set of possibilities. This flexibility will be explored in more detail during the discussion of presentation and interface, but let us first look at the connection between plot and interface.

Interface & plot

Plot is the structural backbone of a narrative. It dictates what events or actions are seen by the viewer and which are implied. It also governs the order in which these events are presented to the viewer. A traditional film provided no influence on the plot. Likewise most video games today provide the player with little, if any, influence on the basic structure of the game. A certain chain of events are lined up and the player moves through those events (stages) one by one as a very linear experience. The only impact the player's action has is whether the game is terminated as the character dies or whether they move on to the next stage. Alternatively, one can imagine a system in the future that would allow you to directly specify what elements and events you would like to have in this narrative before the presentation even starts; turn down the violence knob, turn up the sex knob, and/or ask for a thriller with vampires in it. Earlier nonlinear video system are also examples of direct manipulation of plot. In these system a branching structure required that the user/viewer decide (amongst a set of choices) what footage would be viewed next. (An early example of this is the interactive video disk series *Murder Anyone*.) These interactions are course granularity of input leaving large segment of video to be viewed in-between brief moments of input. This segments the presentation of the story with brief moments of viewer interaction.

Examples of indirect influence of a plot are not so obvious. Take an example where the user is given a knob that can adjust the tension and/or pacing of the narrative they are viewing. By turning this knob up, the need to affect the plot by adding or removing events increases the tension. There are a number of different levels of narrative structure, and at the higher levels the plot may never be affected. For instance, in a love story it might be predetermined that boy meets girl and they fall in love. Regardless of how much input the viewer has this will not change. But at a lower level in the plot the viewer may in fact change how or when they meet. At this lower level the viewer may have a profound influence. The issue of adjusting the tension is complicated by these different levels of narrative structure, because tension is influenced by changes in all these levels of plot as well as by changes in the presentation.

Interface & presentation

In the previous chapter the presentation part of an interactive narrative was broken down into the geometry, its motion, and the camera. The geometry is the part that represents the physical and visual properties of the elements in the narrative. These elements include the characters themselves, the props, the sets, the customs, etc. This geometry is then manipulated over time (animated) to provided the motion. This motion includes both the simple routine, or mechanical movement of background elements, to the subtle movements that make up the performance. The camera, like the geometry and motion, has a location, orientation, and motion, but is treated separately because of its unique relationship to the viewer. All of these presentation elements have the same potential for manipulation that the plot has; direct, indirect, or none. The relationship between the presentation and the interface can be better understood by looking at some examples of camera control.

Head mounted displays are becoming a more and more common method of both display and interface for interactive systems. A head mounted display generally works by attaching a display to the front of the viewer's head. An additional device used to track head movements is also attached. With information about the head orientation and position, the image in the head mounted display can be updated as the head moves. This gives the user the illusion of being immersed in the synthetic world. In other words, the camera has become the eyes of the viewer. This is an example of the interface directly manipulating the camera. The user's inputs (head movements in this case) are immediately and directly used to alter the camera position and orientation.

Recent research has shown the promise of systems that could some-day automatically film an event (Karp and Feiner 1990; Drucker 1994). Whether it is to show a particular operation for instructional purposes, or to film an event in a particular style for artistic reasons, these systems would generate the appropriate camera information. As systems like this come to fruition, interfaces could allow the user to indicate an interest in a particular event or character. This input could then affect the way in which the narrative is filmed, showing more or less of a character or adjusting the manor in which they are to be filmed. In this scenario the user is indirectly changing the presentation.

These examples have treated all the aspects of the camera (position, orientation, field of view, etc.) as a unit. The influence on all the parts have been either direct of indirect. There is no reason that the different parts of the camera can not have a different relationship to the interface. Take the situation where you are riding on a bus. In this case the you have complete control over where you look (the orientation of the camera) while the bus driver is controlling your location (the camera's position). This example servers to show how all the different degrees of freedom available for control by the interface can have a different relationships to the interface, so directly manipulated while other indirectly or not at all. This ability to sperate the different elements provides a rich set of alterative ways of constructing the interface to an interactive narrative.

Voluntary vs. involuntary input

The user sits in front of a screen with a single knob in front of him/her. As the narrative plays out it is adjusted to increase or decrease tension when the viewer turns the knob up or down. Like most of the input examples that have been discussed so far, this example shows the viewer consciously generating the input. To date in most systems the need exists for the user to step back from the narrative and take note of the need to interact and then physically activate an input device. Alternatively, some or all of the inputs to an interactive narrative could be generated by the viewer unconsciously.

Take the same example but instead of using a knob, use a non-intrusive reading of the viewer reaction to the narrative as input. Assume for the sake of this example that how close the viewer sits to the edge of the chair is an indication of how tense they are. (This is most likely not a practical "unconscious" input, but it does serve to explain what is meant by unconscious input.) This new scaler information (distance to edge of seat) can now be used to change the tension. Assume you want to keep the viewer on the edge of their seat. As the viewer sits back in the chair, the tension could be increased. Then as the viewer sits up the tension could be stabilized to hold them at this point. When interactivity is derived from unconscious input, it becomes less of a means by which the viewer controls the experience and more of a tool by which the author can control the audience. Another example might be that a narrative would have a graph of the tension level over time. The narrative could then be continuously altered to match the tension level in the viewer with the current level dictated by the graph. There are exercise bikes currently on the market that alter the level of difficulty by monitoring your pulse rate as you ride.

The viewer, the camera, and the presentation

There are two more qualities that help categorize interactive narrative that have yet to be discussed. These qualities clarify the viewer's relationship to the narrative and the camera's relationship to the elements (the geometry) of the narrative. The technology of interactive narrative is uniquely different from traditional form of narrative in its ability to cast the viewer in the role of one of the characters. Being in this role influences the relationship between the viewer and all the parts of the narrative including, the plot, the other characters, the setting, the props, etc. But probably most important, this put the viewer/user in the situation where the other characters in the narrative are aware of his/her presence. This relationship allows these characters freedom to directly react to and talk to the viewer showing a cognizance of the viewers. Alternatively, the viewer can have a more traditional relationship to the narrative that sets them outside of the narrative in a god or ghost like presence. In this role the viewer looks on the action without the characters in the narrative being aware of his/her presence. This does not mean the viewer does not have influence on the narrative, but that influence (regardless of whether it is direct or indirect) happens without the characters being aware of the viewer. A ghost or god can have an influence on the world without their presence being known. Regardless of whether the viewer is god or not, the type of influence the viewer can exercise is left to be specified. Likewise the camera's relationship to the narrative is left open.

The camera is unique in that it is the means by which the viewer visually perceives the narrative world. Because of this special attribute, the camera has a relationship to the narrative world all its own. A camera can hang disembodied and invisible observing the narrative, or the camera can reside in the head of a character looking out of the eyes seeing only what that character sees. As the example at the beginning of this chapter illustrates the camera can be outside of the characters looking on the action, while the viewer/player for all intents and purposes is the character (Mario, the HedgeHog). This makes the camera's relationship to the narrative a characteristic that is independent of all the other characteristics discussed.

These variables of interactive narrative like the others discussed are ones that can change throughout the course of the narrative presentation. That is to say that there is nothing to prevent the viewer from playing the role of one of the characters in the narrative with the ability to directly manipulate the environment during one scene, then having his/her role transformed so that they are a god-like onlooker with only indirect influence during the next scene. While these variables help us to discuss and classify interactive narrative, they do not necessary hold the same value for an entire narrative.

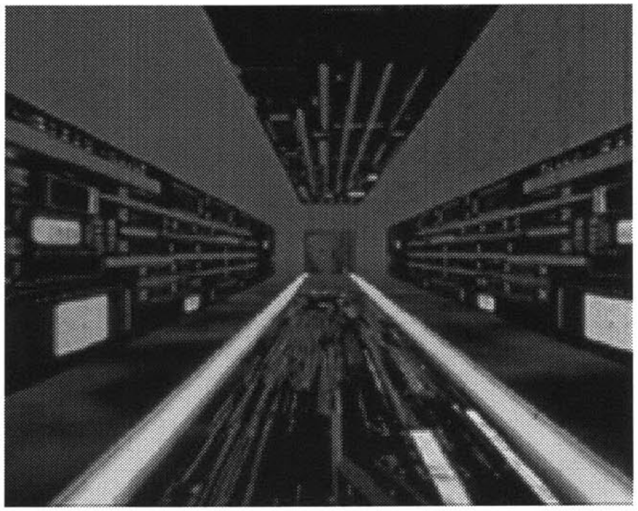
Chapter Seven

First Experiment

This first experiment was produced to be a permit exhibit at the Chicago Museum of Science and Industry. The constraints of bringing real-time computer graphics, coupled with immersive interface technology, to an exhibit provided a good opportunity to test some of the ideas behind the theory of narrative guidance of interactivity.

This piece is the highlight of the Museum's new exhibit, *Imaging the Tools of Science*. The primary goal of this exhibit was to expose and educate the visitor to what "virtual reality" technology is and what it can do. Any experience that was going to be successful, was going to be highly constrained by the issues inherent in bringing an immersive experience to a public place like the museum. In a museum setting it is necessary to limit the amount of time a person spends, provide an interface that keeps people from getting lost and frustrated, while at the same time making them aware that they have some direct and immediate control over how they move

The virtual version of the museum space. Stage1.



The outer space like virtual video camera space. Stage 2.

through the environment. To meet these demands it was decided that the experience would be between 2 and 3 minutes long with a clear beginning, middle, and end. This allowed the user to feel they had a complete experience while allowing the museum to predict how quickly they could move people through the exhibit. These constraints required the user's navigation to be guided through the virtual world, and the river analogy helped address these issues.

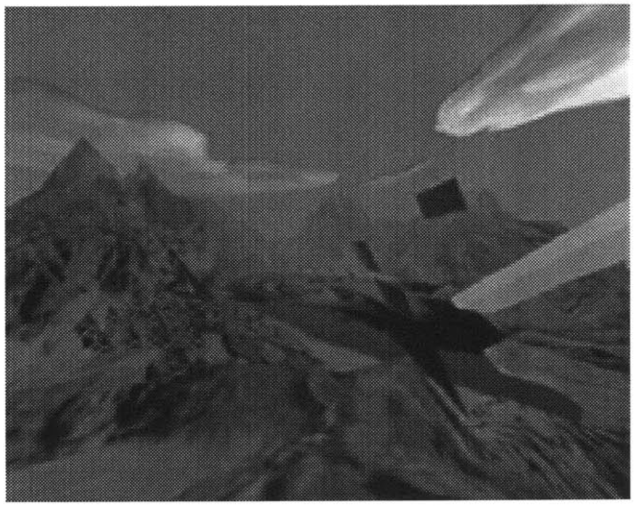
In this application, the analogy of the river was taken quite literally. We defined a path through the virtual space as the river. The user was then guided through the space much like a water-skier behind an invisible boat. This approach allows the user to continuously interact with the presentation while also being loosely and sometimes not so loosely constrained within both space and time.

Content

In addition to entertaining the audience, the experience needed to fit thematically in the context of the entire imaging exhibit and fulfill the educational goals set forth. More specifically, the user should learn through experience that a virtual environment can be anything, can changed at any time, and is responsive to you and your presence in it.

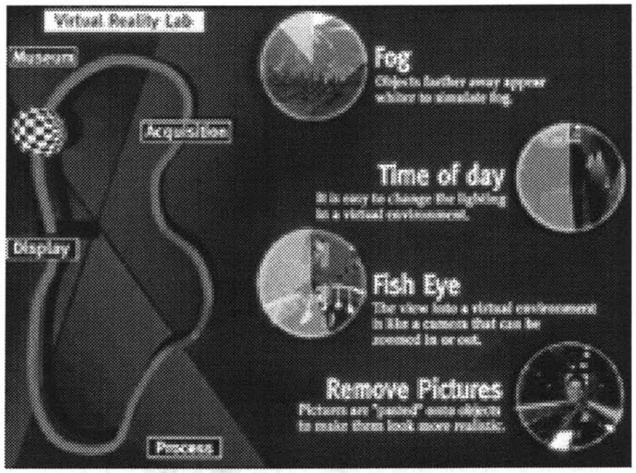
An experience that represented the "image processing story/journey" was chosen. Image processing consists of three stages: acquisition, processing, and then display. By taking the user on this journey a story with a beginning, middle, and end can be told while accommodating all of the goals and requirements listed above. The viewer flows, navigates, and is pulled through a symbolic representation of image processing. In the way that a video images, travels from the museum space through a video camera into a computer to be processed, and then back to a CRT for display, the user flows through the following four stages. First, is a virtual model of the museum space in which the viewer is currently. Second, the user follows a floating human face as it enters the surreal outerspace-like insides of a video camera. They then ride through a coax cable before entering a city. In this city, images of faces on billboards are processed before moving on into a natural hillside environment. In this last section a image is scanned on to the surface of a lake like the e-beam of a CRT scans an image onto the surface of a screen. The journey ends as the user splashes through the lake to return to the virtual museum space.


The virtual processing city. Stage 3.

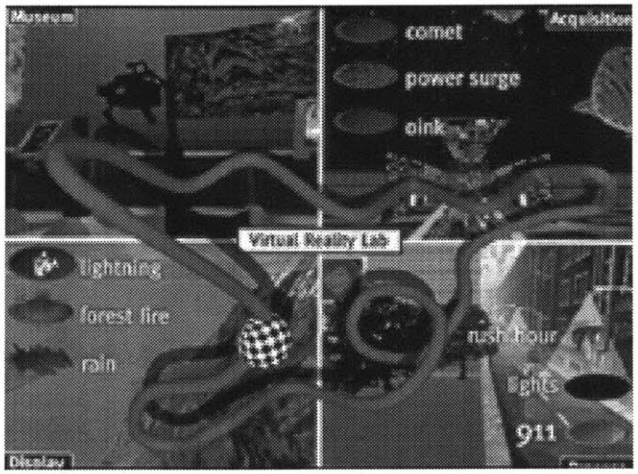

The mountainous virtual display space. Stage 4.

Setup and equipment

The Virtual Reality Laboratory is meant to introduce the public to virtual reality or virtual environment technology. Because of this both the museum and the developers felt that a key element was immersion. To create a sense of immersion both a mobile display device (typically head mounted or boom mounted) and device to track head movements had to be chosen. A Fakespace boom2C was used to address both of these issues. The boom provides a stereo pair image from two CRTs. These displays are mounted on a counter balanced armature. This armature also serves to track the users head movements. The boom, coupled with a Silicon Graphics 4D/440 Reality Engine, provided the ability to generate highly textured mapped images in real-time. A MIDI sampler is triggered from the SGI to provide both ambient and spatialized sound during the experience. The ambient sound is wired to speakers in the room for all to hear while additional sounds are spatially located (by a Crystal River Beachtron) and played through two small speakers mounted on the boom. One of the images generated for the boom is repeated



First Macintosh touch screen. Allowing people to trigger virtual events. For example changing the time of day.



Second Macintosh touch screen. Allowing a third person to trigger additional environment specific events. For example "lightning."

for everyone to see on a large screen.

Because of the prohibitive cost of this equipment it was unpractical to have more than one immersion station. But it was desirable to have more than one person actively involved in the experience at any one time. The solution was to add two Macintoshs to the network. These displays provide buttons through which people, external to the experience, can trigger events. An iconic map of the person's journey (with an indicator for their current location) is also displayed on the Macintosh screens (figure -- picture). By involving more people the educational goals of the project could be better met.

Interface

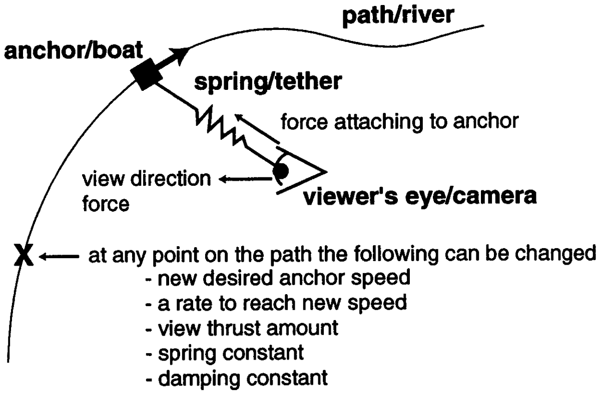
The user interface solution is one that gives the user complete control over the direction of the view. Just like with most immersive environments, the head movements are tracked and when the user looks left the image is updated so as to have them look left within the virtual environment. The user's position in the virtual space, on the other hand, is only partly under the user's control. Local movements are a product of the user's actions while the general area that the user is in at any given time is predetermined by the author/ director.

The analogy that best describes this user interface is the one mentioned previously. That is a water skier behind an invisible boat. The boat or anchor moves along the path at a rate that varies as specified by the creator of the experience (the author). Users are then tethered to the anchor by a spring that constantly pulls them along. Meanwhile the user is free to look in any direction he or she chooses. The figure shows the model used. This model gives the user direct control over where s/he is looking while at the same time giving him/

her indirect control over their local position. Looking in a given direction will impart some force in that direction and allows the user to swing over in that direction moving closer to the object being watched. At the same time the boat continues to pull him/her along the journey, maintaining a sense of pacing and flow.

There are a series of parameters that can change the nature of this interface: the current and desired speed of the anchor, the amount of thrust the user is imparting, and the spring and damping constants. In this implementation, all of these values are free to change throughout the experience. The changes are encoded at locations along the path, allowing the author to specify over which areas of the journey the user is more or less free to roam. For example, as the user approaches a larger open space the author may choose to slow down the anchor, decrease the spring and damping constants, and increase the viewer thrust allowing the user more latitude and time to explore. Alternatively, the author might focus the experience by increasing the spring constant, speeding up the anchor, and reducing the thrust.

The path is also used as a way to trigger events. Triggers are encoded on the path and as the user passes those points they are fired. For example, there is an animated pixel swarm in the experience. This pixel swarm starts as an image, comes apart and flies off, and then later coalesces in a new location. This swarm is often triggered to fly off or coalesce as the user passes. This is a simple way to indirectly link the events in the virtual space with the actions of the user. As mention before other events and sound are also triggered from the two external MAC interfaces.



An application of the River Analogy, consisting of a number of different parts: the anchor moving along the path, a spring attaching the user position to the anchor, a thrust imparted by the user dictated by the direction the user is looking, and a general viscous damping to prevent the user from oscillating about the anchor position.

Guidance

The goal of a guidance system is to allow the user to have continuous and smooth interaction while also being constrained to an interesting and compelling path. The interface described above was designed to do just that. This interface by its nature mixes the ability for the user to have complete control over parts of the interaction while constraining him/her to traveling particular paths. The interface is the primary way guidance is achieved. But there are a couple of additional guidance issues that are not addressed by the interface as described.

This experience is divided into four sections. There is a transition between each of these sections. The first transition moves the user from the virtual museum space into a virtual video camera. This is achieved by taking the user through the front of the lens. But this transition is lost and is not understood because the user does not see the approaching camera lens. When the viewer has complete control over the direction of view they can miss key story elements. The solution was to pivot the user so that they would be facing the correct direction. You can think of this as riding in a car and no matter which way you turn your head the seat turns to assure that you are looking forward. Once the transition has been navigated, the control is released and the user is free to look where s/he wants. In this dynamic environment, where the user is constantly traveling, this momentary loss of control seems to be acceptable. In most cases it goes unnoticed.

So far this flow has been talked about as only one path. In fact, the path in this piece has forks. These forks are places where the anchor, behind which the user is pulled, can switch tracks. This allows the user to go left, right, up or down. The decision as to which direction the user will go is made based on the direction that s/he is looking when s/he approaches the fork. If the user is looking off to the right, s/he will take the path that moves off to the right. The idea being that if s/he is looking in a particular direction s/he is most likely interested in what is over there. These forks are not visible to the user. Therefore it is only after repeated use that a visitor becomes aware of them.

This interface technique is easy and interesting for all to use. It serves the museum environment by providing the author/director with enough control over the experience while retaining both the interest of the viewer using the boom and the audience watching on the repeater screen.

Plot & presentation

To link this back to the architecture described in the approach chapter we need to look at the representations of plot and presentation. This piece is structured as a journey. During this journey the presentation has a temporal flow, events unfold to provide a sense of pacing. The interaction allows the user to continuously adjust his/her relationship to the virtual environment. But this journey through a surreal world has little or no plot. There is a beginning, middle and end, but there are no characters, no character development, nor any story being told. Temporal structure is one aspect of plot, but unlike a good film, this piece does not have character development or causality.

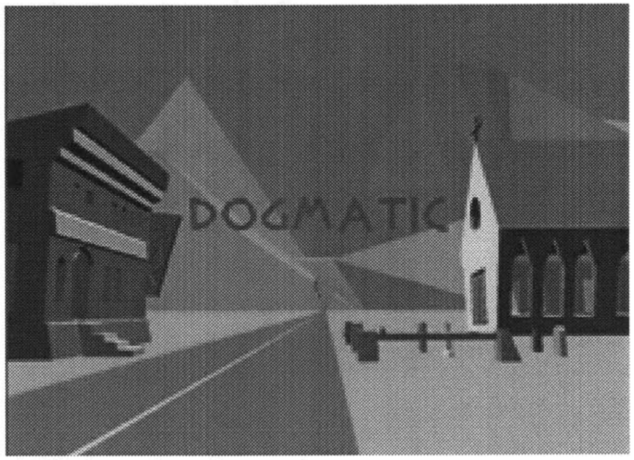
On the other hand there is a clear presentation level. The spatialized sound, three dimensional models, and simple animated elements all create a presentation.This presentation level is smoothly and continuously manipulated by the user via the immersive interface.

The temporal flow and the continuous presentation level interaction fulfills two of the four principles I outlined in chapter two. But the other two principles are only partly met. There is no computational representation of plot. Also the use of staging and transitions is simple. Without a more sophisticated use of staging and transitions, there is no way for a plot to manipulate the time and space of the presentation. The simple, animated elements are insufficient to be called characters. Without the ability to direct their behavior, the staging can not change to account for the user's behavior. These problems are addressed in the second piece, *Dogmatic*.

Chapter Eight

Dogmatic

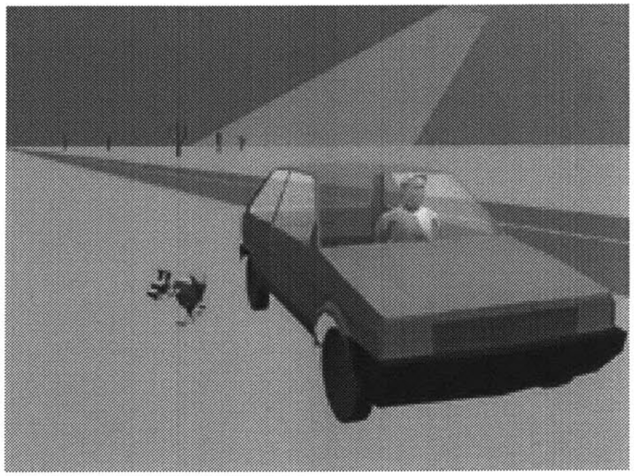
Dogmatic is the title of the second experiment in narrative guidance of interactivity. It answers to the shortcomings of the first experiment outlined in the previous chapter. This piece was constructed to tell a particular story. New tools where developed in order to bring the power of storytelling techniques to bear on virtual environments. As the narrative guidance of interactivity approach suggests, *Dogmatic* is authored to assure plot points are met while allowing the user to continuously interact with the presentation.



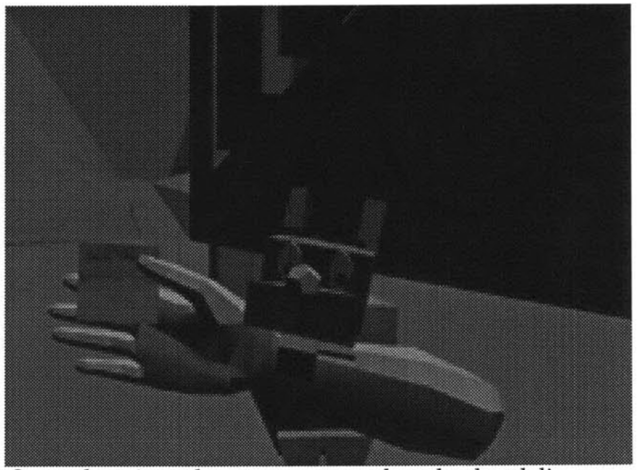
A title frame from the opening sequence.

The story

Dogmatic is a short story (under ten minutes) in the Noir style. It is set in a playful, cartoonish southwestern dessert. Mountains in the background provide an overall orientation. The action takes place in the vicinity of crossroads, where there are few buildings and some



Scene two introduces the dog as the main character.



Scene three introduces a mystery when the dog delivers a arm with a note in the hand.

randomly placed cacti. The buildings provide both a context and a means for the user to establish his/her orientation throughout the experience. The user is cast as a character in the story. Additionally, there is a dog (the main character), a car and its driver, and a different car (only briefly encountered).

The story consists of six scenes each with a particular goal or point to convey. They include:

- **opening:** This scene serves to establish setting and allows the user to become comfortable with the interaction. It also serves as the title sequence.
- **car & dog:** This scene serves to introduce the dog as the main character, and to establish its relationship to the driver of the car.
- **find the note:** This scene disrupts the equilibrium and introduces a mystery. What do the words "Lucky Strikes" on the note mean and why was is it in the hand of a disembodied arm delivered by the dog.
- **night:** This scene builds the mystery and serves as a link between the first and second half of the story.
- **car accident:** In this scene the dog returns and appears to inadvertently cause you to be hit by a car.
- **closing:** You come to and find the dog standing over you. You then learn that the name on the dog's tag is "Lucky" right before he walks off with your now dismembered arm.

The story takes place over approximately a 24 hour period. While transitions are sometimes used within scenes to foreshorten time, the large time ellipses are between scenes. This is done mainly with cross dissolves. Changing orientation, lighting and sometimes position of the user during the cross dissolve is an effective way to convey the passage of time.

The pacing of the story fluctuates over the course of the story. It peaks in the second and fifth scenes. It is interesting to note that these scenes employ the most cuts. The introduction of faster and more frequent cutting is a powerful tool for increasing the pacing.

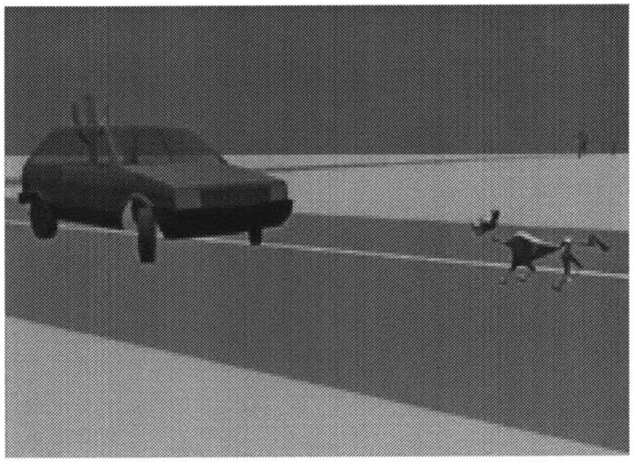
Setup & equipment

This piece, like the first, was developed as an immersive virtual environment. The software and the models are run and displayed on a Silicon Graphics workstation. There are two different interface versions. One that uses the mouse and another that uses a head-mounted display. A VR4 head-mounted display from Virtual Research was used. It was coupled with a Polhemus FastTrack to provide the head tracking.

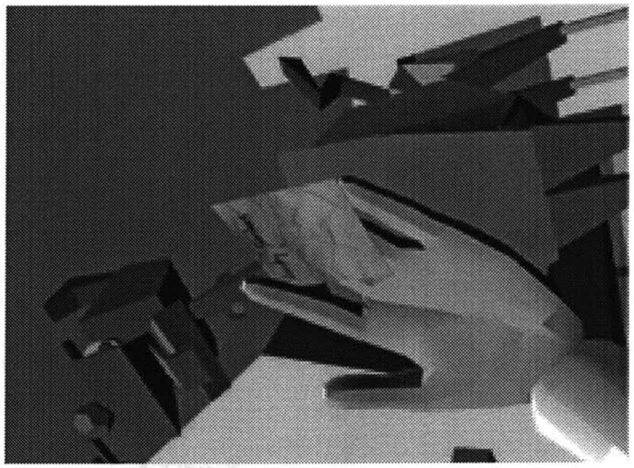
There are two sources of sound in the system. First is source is the sound effects or foley. These noises are associated with objects in the world and their sound is spatially located with respect to the user. For that reason the user wears headphones. These spatially located sounds have proved to be very effective at getting the user to look in a particular direction. The spatial location is done in software by simply adjusting the left to right balance.

Music is the second source of sound. In an interactive system like this one the duration of an event or scene cannot be predetermined. This means that a flexible music system must be employed. For this work the DBX system developed by Alex Rigopulos, Eran Egozy,

A night scene that functions to heighten the mystery and connect the first and second halves of the story.



Scene four, a car serves to miss the dog and hits the user.



In the closing scene the user comes to see the dog sanding over him/her. Then the dog walks off with their arm.

and Damon Horowitz was used (Rigopulos 1992). Teresa Marrin composed new music specifically for *Dogmatic*. The DBX system is a generative music system.The music is divided into clusters. Each cluster consists of three parts: a looping background sequence and two generated textures. These textures can be transformed to dynamically change their character. A number of different clusters are used throughout the piece and are regularly adjusted. The DBX system runs on a Macintosh and generates Midi for a synthesizer.

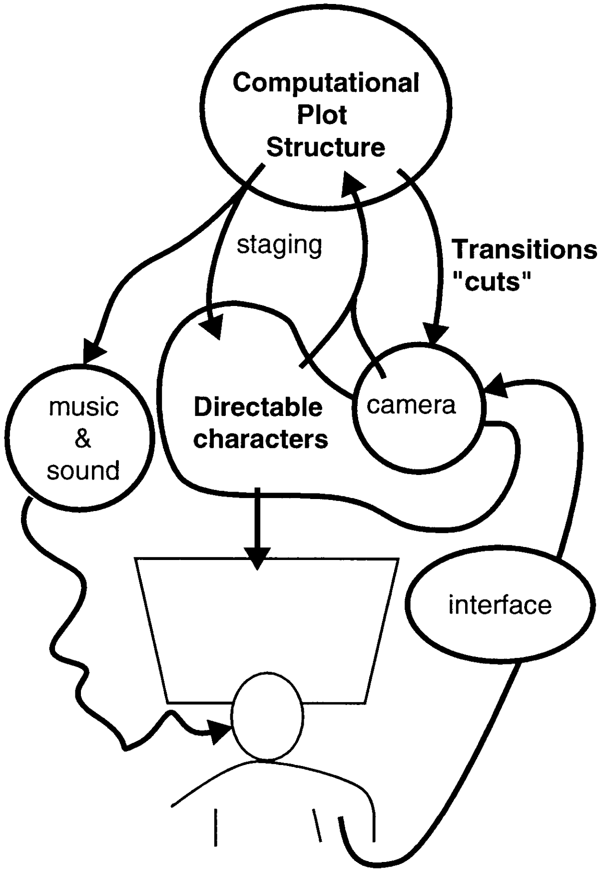
Interface

The interface for *Dogmatic* is simple. Just as in the first experiment the user controls the direction of view. However, in the first experiment this head tracking was coupled with a software implementation of the water-skier. The interface software provided the guidance. In *Dogmatic* the guidance properties of the system are separated out from the interface. The computational plot is responsible for using staging and transitions to affect this guidance. Therefore the interface is simply the hardware and software that allows the user to control the direction of view. Two methods were experimented with, a mouse and a head-mounted display coupled with a tracking device.

The mouse interface is the more straight forward of the two. Only the mouse movement is used (buttons do nothing). Moving the mouse left, right, up, and down tilts and pans the camera (your head). The second interface used was a head-mounted display coupled with a 6-degree-of-freedom tracker. Only the three rotational degrees of freedom were used. This interface provided a more natural kinesthetic link between user and camera as well as a greater sense of immersion. But this interface also makes cutting more difficult.

These interfaces give direct control over the camera, but the system also has to exercise control over the camera when it introduces cuts. A cut re-positions and/or re-orients the camera. There is a base frame for the camera (often the user's body). This base frame has its own position and orientation. The user then adjusts the camera's orientation with respect to this base frame (in other words controls the turning of the head). When the mouse cursor is in the center of the screen the user is looking in the same direction as the base frame. Therefore any offset from center is an offset in orientation. When a cut occurs the base frame needs to be repositioned and/or oriented to ensure the user is looking in the correct direction. To do this with the mouse interface the base frame is set with the correct orientation and then the mouse is warped back to the center of the screen. This ensures the user will be looking in the correct direction while also resetting the mouse to be centered with respect to the new base frame.

This problem is not so easily solved when using a head-mounted display. This is because the user's head cannot be warped back to center. First implementations tried to get by without re-centering. As long as most head movement is side to side and the user is free to turn all the way around, this works pretty well. Another approach is to coordinate the cutting so that any offset introduced by a cut is later reduced by another cut. This is very difficult to manage and is often too restrictive. This suggests it is necessary to re-center the user's head. If re-centering does not happen the user will quickly find herself/himself needing to hold an unnatural position just to look forward. The next approach taken is to record the rotational offset needed to re-center the user's head after a cut. This offset is then slowly reduced over the next couple of seconds and the system



Repeat of diagram from chapter two (Approach). This is the architecture used in *Dogmatic*.

relies on the user to turn his/her head to compensate.

Plot guidance of presentation

In *Dogmatic* the guidance comes from the plot level. The plot guides the presentation. This is accomplished by the architecture outlined in the approach chapter. The diagram is repeated here. There is a computational plot model. It uses an author specified plot structure to affect changes in the staging and transitions. In *Dogmatic* the structure is authored to assure narrative integrity, i.e. to assure the story is told. At the presentation level the world is primarily represented as directable characters. These characters, along with the static sets, populate the world. All of these elements at the presentation level (music, sound, and directable characters) are manipulated by the plot to provide the narration. It is important to note that while the research does not focus on the use of sound, it is an essential element of the presentation. The staging of most events is accompanied with sound and all off screen events are staged exclusively with sound.

The remainder of this thesis focuses on three parts of the architecture: plot, characters, and transitions. A chapter for each follows.

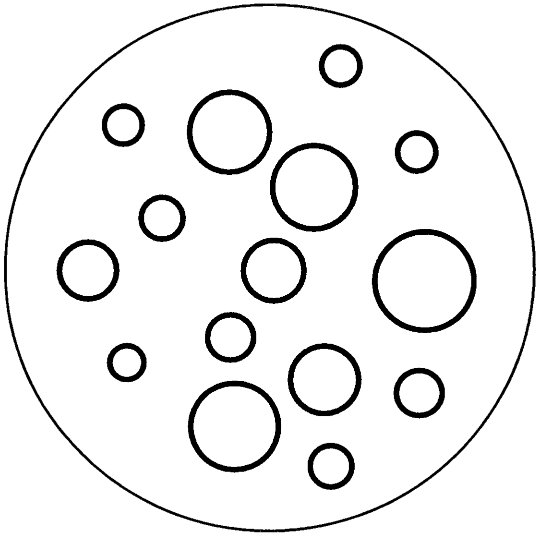
Chapter Nine

A Computational Plot Structure

The problem of event selection

A story is told by stringing together a groups of events. Together these events serve to unfold the key elements of the plot, i.e. plot points. If done successfully these plot points are unveiled over time to weave a compelling presentation. This thinking leads to the separation of the narrative into two parts; the events themselves and the presentation that delivers them to the audience.

Having made this distinction the problem of computationally representing plot can be thought of as an "event selection" problem. A pool of events are provided by the author, but these events must be presented to the user. What event do we present next and when do we present it? The problem is to evaluate the current situation and decide what is the most appropriate event to pull out of the pool next. The first step is to decide what criteria influences the decision.



The Event Selection problem is one of deciding what event to take from the pool next and when to execute.

It is important to note that this work is intended to provide an author with the ability to create and represent a reactive plot. That is, a plot that can adjust and react to the variation in the environment that comes as a product of user interaction. This is quite different from the process of dynamically building a plot based on this interaction. The goal is to provide tools for the author to better tell his/her story, not to provide the users with the ability to construct their own story.

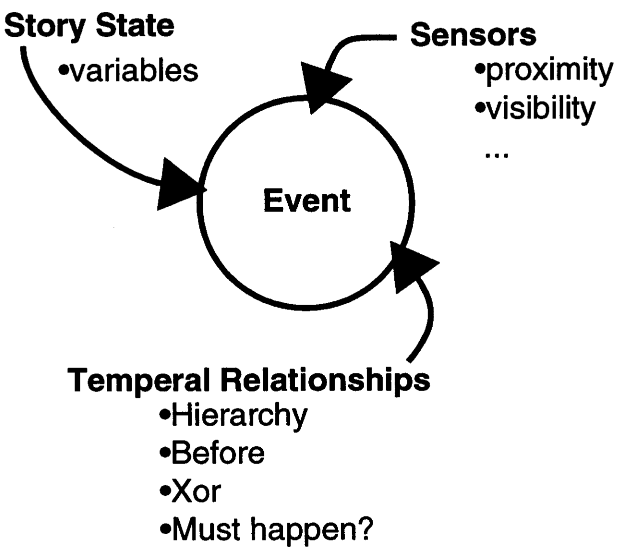
An event

Instead of looking at the pool of events as a whole, let us look at the events within the pool individually. The event selection process is then one of evaluating whether a given event is suitable to be chosen for the current moment. If we can do this we can then evaluate the fitness of all events individually and determine which events, if any, are currently acceptable. I have chosen three criteria to make this judgment: story state information, sensory input, and predefined temporal relationships amongst events.

Story state

Throughout its presentation, certain aspects of the story are constantly changing. The pacing of the story and the viewer's awareness of a particular character are both examples of story state information that could be represented as variables. These high level story state variables may play an important role in deciding what event to choose next.

Say that at a particular moment in the story it is important for the user to become aware of a particular character, the dog. This is a goal which the event selection algorithm must accommodate. As the user freely pans the landscape, the dog comes into the frame. Cutting to a close-up would raise the user's awareness of the dog. Therefore this close-up event would be an appropriate choice. An event's relationship to the story's state can be encoded with the help of a story variable; a variable that represented the current level of "dog awareness." The close-up event can then note that it is an event which will increase the "dog awareness" variable.



There are three things that influence the selection of an event. They are the current state of the story, the state of the environment (as seen through the sensors), and the event's temporal relationship to other events. An example of how events are scripted can be found at the end of this chapter.

Sensors

Sensors are computational elements that look into the world and provide information about the current conditions. For example, a simple sensor might report whether or not the dog is visible to the user, or it might give a distance between the user and the dog.

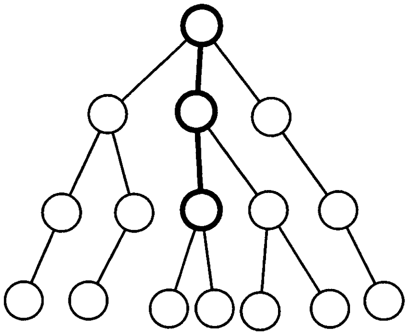
Clearly these sensors can aid in the decision of an event's current fitness. Using the same example, it may only be appropriate to cut to the close-up of the dog once the dog has been seen. Therefore a given event may have a link to any sensors necessary to evaluate its fitness.

Temporal relationships

The temporal relationships are the most extensive of the criteria for deciding an event's fitness. They are also one of the main things that differentiate this "event selection" algorithm from other "action selection" algorithms. Any set of story events has a temporal relationship to each other. An event can be large and encompass many other events. Or an event might have to happen before another. In another case two events might be mutually exclusive. These relationships must not only be encoded but also filtered into the selection process.

This encoding is done in four ways. First a hierarchy of events is specified. This allows the author to specify the narrative in a top down fashion. A high level event, like "introduce the dog," can contain a number of events which are free to be combined to suit the current situation. Second and third are the ways relationships among sibling in the hierarchy are formed. Any two siblings A and B are allowed to have one of two relationships. A comes before B (thereby implying that B comes after A.) A xor'ed with B, meaning that A or B can happen but not both. The forth and last piece of information an event must specify is whether it "must happen" or not. This information is represented in what I will call the "must happen" flag. From here on I will refer to any events with this flag set to true as the "must happen events."

Together these four type of relationships place temporal constraints on the order in which the events can be presented.



Events are organized into a hierarchy. A node higher in the tree is an event that temporally encompasses the events below it. With in the hierarchy only one path from the root to the current event is active at any given time (indicated in bold). Siblings have there own temporal relationships outlined in the next figure.

The event selection algorithm

With each time step (in my application with each rendered frame) the system evaluates every event. The first test establishes which of these events are even eligible for selection. Additional information is then used to select one event from all those that are eligible. Unlike some action selection algorithms, it is not necessary for an event to be selected within each time step. The pauses between event selections are just as important as the events themselves. It is only with these pauses that properties like "pacing" can be controlled.

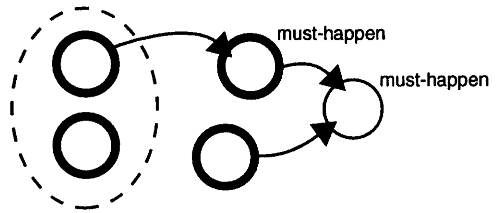
Hierarchy

The events are placed by the author into a hierarchy. A hierarchy was chosen for a couple of reasons. First is that this structure lends itself well to story representation. The root contains the entire story, and its children represent each act, while each act's children contain their respective scenes, and so on. Whether it be chapters, acts, or plot points, stories seem to naturally be organized hierarchically. The second reason for choosing a hierarchy is because hierarchical data structures are a common and well understood computational tool.

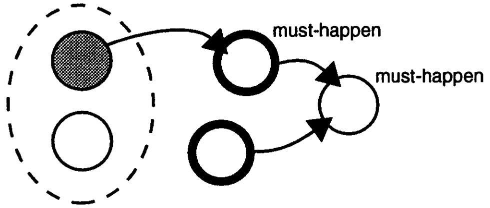
The algorithm starts at the top of the hierarchial "tree" and proceeds down the tree. Once a particular node/event is selected, only its children become eligible for selection. This remains the case until either a specified end condition is met or all of its children have been executed. Strictly speaking, only the "must happen" children (those with their must happen flag set to true) need to have run. Likewise when a child is selected the algorithm is recursively applied until a leaf event is reached. This hierarchy greatly restricts the number of events that need to be evaluated with each time step. Only those amongst the current event's children need be considered.

Relationship between siblings

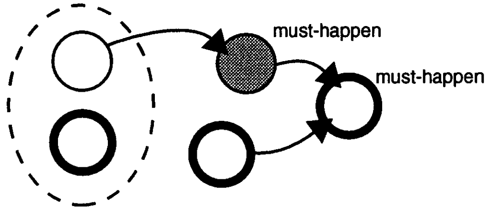
The sibling relationships are the second things that dictate an event's eligibility. There are two of these: "A before B" and "A xor B." The "A before B" constraint implies that B cannot happen until A has and that A cannot happen after B has been selected. Alternatively, the "A xor B" constraint groups A and B together. Each of the events remain eligible until one is chosen, at which point the other is removed from eligibility permanently. If a third event C is xor with either A or B it is, in affect, added to the group, meaning that the act of choosing any of the three removes the other two. Note that under these conditions it is only makes sense to specify one "must happen" flag for the group as a whole. The evaluation is done as follows:



Sibling relations are of two types: A before B (arrow) and A xor B (dotted circle). At any given time only certain events are eligible for selection (bold). Note that the events that "must happen" are indicated.



Here one event has been executed (indicated as gray). Note the event with which it has an XOR relationship is not eligible any more.



In this example a different event has been chosen. Note that the only way the event on the far right can become eligible is if the gray event has executed.

- A before B: In this case event A is considered to be up stream of event B, and likewise B down stream from A. If no "must happen" events that have not been executed are up stream and nothing down stream has been executed, then the event is eligible.
- A xor B: All events xor'ed together are considered a group. If no other member of the group has executed then the event is eligible.

Sensors & state variables

Each event has a list of relationships to both the sensors and the state variables. These links are specified by giving three numbers a minimum value, a maximum value, and an optimal value. By having a link to a particular sensor or variable, an event is establishing a requirement. The requirement is that the corresponding value (sensor or variable) be between the given minimum and max values. The optimal value (which must lie between the minimum and max) is used to create a weight or fitness value. The range minimum to opt and the range opt to max are both normalized and a fitness of 1.0 is given when the value equals opt. The fitness then falls off to 0.0 as it moves toward either the minimum or the max. For a given event which is executable, all of its fitness values are averaged to arrive at a single metric. The event with the largest fitness is then chosen. In the off chance that more than one event has the highest fitness one is chosen randomly.

Executing an event

Once an event is chosen there is the issue of executing that event. In the algorithm presented here only one path down the hierarchy tree is active at a time. This means that at any moment at most only one event is active at any level in the tree. When a leaf event is executed it must be completed before we move back up the tree. This requires that we be able to test if a given event is finished. There are four tests that are employed in this algorithm.

- A callback function is provided by the author. This function is called with each time step until it concludes that the event is to stop. This can be used for both leaf and non-leaf events.
- For leaf events a specified time limit is reached.
- For leaf events it runs until another event is chosen.
- For a non-leaf event it runs until all "must happen" children have executed.

The test used is specified by the author. In my implementation, an event specifies two callback functions. One that is called to start the event's execution and another to stop it. In addition is the callback function mentioned above, that can be used to test for the conclusion of an event. This callback also provides an event with the opportunity to perform a task with each time step.

The control loop

The control loop is responsible for traversing the event hierarchy, selecting events, and executing them. It is outlined in the pseudocode.

```
currentEvent = root
currentEventNoWait = NULL

loop for each time step {
    update all sensors

    while (currentEvent->isDone()) {
        currentEvent->stopExecution()
        currentEvent = currentEvent->parent
        if (currentEvent == ROOT)
            stop
    }

    if (c = currentEvent->pickChild()) {

        if (currentEventNoWait) {
            currentEventNoWait->stopExecution()
            currentEventNoWait = NULL
        }

        c->startExecution()

        if (c->isNoWait())
            currentEvent = c;
        else
            currentEventNoWait = c;
    }
}
```

This pseudo-code outlines the main loop of the event selection algorithm.

Authoring an event hierarchy

This model of computational plot in no way diminishes the role of the author. It was developed to be a tool for the author. So how does one go about authoring such a event hierarchy? The first step is to

```
creature car <car-file>
creature camera <camera-file>

sensorVisibility seecar camera car
sensorProximity nearcar camera car

event example
ecd example
    event car-approaches
    eventSetCBs car-approaches {car position X Y Z;
                                car move SPEED} {} {}

    event car-honk
    eventMustHappen car-honk 0
    eventSensorLink car-honk seecar -1 -1 0
    eventSetCBs car-honk {car sound honk} {} {}

    event close-up
    eventSensorLink close-up seecar 0 1 1
    eventSensorLink close-up nearcar 1000 1000 500
    eventSetCBs close-up { ... } {} { ... }
    ecd close-up
        ...
    ecd ..

    event pull-over
    eventSensorLink pull-over nearcar 0 500 500
    eventSetCBs pull-over { ... } {} { ... }

    eventAbeforeB car-approach car-honk
    eventAbeforeB car-honk close-up
    eventAbeforeB car-honk pull-over
    eventAxorB close-up pull-over
ecd ..
```

This is an example file that specifies an event hierarchy. The characters, sensors, and story state variables are created at the top. The events are then created in a hierarchy, only one level deep in this example. All sensor links and sibling relationships are specified.

specify the characters that will be used in the story. Second is to specify the sensors and the story state variables. With these in place the individual events can be specified.

Each event is created within its location in the hierarchy. Once created, all links to sensors and state variables are specified. For each link the name of the sensor or variable is given and the three (minimum, optimal and maximum) numbers are given. The next step is to write the callbacks for the events (the start, end, and during callbacks.) I have found that I rarely use the "during" callback. This is the callback that is called for every time step the event is active. By default an event's "must happen" flag is true and an event's duration is specified to be of length zero, but is easily changed if the author desires.

For each group of siblings the before and xor relationships need to be specified. If no relationship is given then all the siblings are considered to have no temporal restraints with respect to each other. Therefore all become eligible (pending sensor and variable links) for execution once their parent is activated.

In this implementation a hierarchy is specified via an interrupted front end. This front end provides a set of commands for loading characters, creating sensors and story variables, and building the event hierarchy. The diagram gives a simple example. Note the "ecd" command is used to move up and down the event hierarchy. It is like the Unix "cd" command for moving throughout the file system hierarchy. In this example only two characters are loaded: a car and a camera. Two sensors are created, one testing the car's visibility and the other the proximity between the car and the camera. Four events comprise the script: the car approaching, the car honking its horn, a close-up of the car, and the car pulling over. The sensor links and the before and xor relationships assure the following:

- The car will always start approaching before anything else happens.
- The car will only honk if the camera is not looking at the car.
- If the camera looks at the car before it gets too close, the close-up of the car will happen.
- The car will pull over only if the close up did not happen.

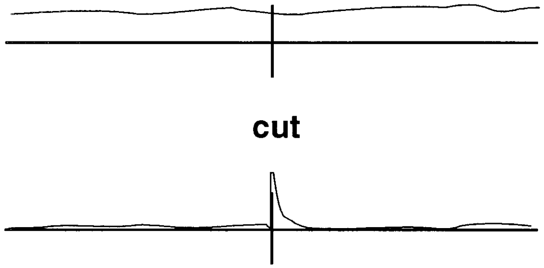
This is a simple script that might be used to assure an introduction of the car to the user. The car honks in an attempt to get the user's attention. If s/he then looks at the car a close-up happens and the goal is attained. Otherwise the car is instructed to pull over and the story can employ other techniques to bring the car to the user's attention.

Chapter Ten

Editing While Immersed

Over the decades since its birth, modern cinema has continued to evolve a language for cuts. Cuts can tell us who is bad and who is good, they can assure we see an important object or action, and they can control the pacing or tension in the presentation. Can this powerful technique be coupled with immersive interface technology?

Immersive interfaces allow the user to see and interact with the virtual environment. It is for the heightened sensation of "being there" that these systems are most known. The technology provides the user with a direct, intuitive link to the virtual environment and mimics the way we interact with the real world. Cinema's power is its ability to do things that we cannot in the real world like transport us in space and time. Likewise, interactive immersive environments should also have the ability to manipulate space and time. But how can cuts be used in a situation where the user has control over the direction and maybe also the position of the camera? This chapter looks at this basic problem, bringing background knowledge from cinema and applying it to a number of scenarios in an immersive experience.



Above are two time-lines each with a cut in the middle. The first is representative of a film where the author (filmmaker) as complete control. In the second example, an immersive interface removes the author's control until the cuts happens when control is momentarily regained.

Before and after the cut

A cut is an instantaneous change in the visual (and sometimes also audio) image. The success of a cut relies on the ability of the viewer to understand this change from old to new image. The new image appears in the context of the image that proceeds it, and relies on this context to be understood. It is control over these images that allows the filmmaker to create understandable cuts. Most often the goal is to have viewers be transported by the cut without making them aware of it. For the viewers to become aware of the cut would mean that they are removed from the story. The filmmaker has complete control over what is in the frame both before and after the cut. They can manipulate the camera and the objects in front of it ensuring that the cut will be understood. In an immersive environment we do not have this luxury.

In my work the user is given control over the direction of the camera as if it were his or her own eyes. In this case the author has little control over what is in the frame at any given time. There is no guarantee, for example, that the user is even looking in the right direction to make the cut work. While there may be no control over the image that is cut from, there is complete control over what image is cut to. This control is only momentary because the interface allows the user to quickly depart from this orientation. So how can the author assure that an understandable cut is made? The approach I have taken involves two strategies:

- monitoring the user waiting for acceptable conditions

- adjusting the staging of the presentation to promote acceptable conditions.

Once the conditions are right the cut is made and an appropriate new position in space and time is provided.

Spatial and temporal articulation

In his paper on spatial and temporal articulation for film, Burch provides a structure for classifying cuts. These are outlined in the diagram. They include five temporal and three spatial articulations. This articulation allows any cut to be classified in one of fifteen ways. While this structure provides a terminology for talking about cuts and a method for analyzing them, it does nothing to explain what motivates a cut. It does not answer the question of why this cut at this time, or why one cut "works" while another does not. At the end of his article Burch touches on the notion of a match cut. Here he begins to explore what makes a cut understandable.

Continuity cutting

On the other hand, Reisz and Millar specifically outline what criteria must be met for a transition to provide a lucid continuity, or smoothness. They are:

- matching consecutive actions
- extent of changing an image size and angle
- preserving a sense of direction
- preserving a clear sense of continuity
- matching tone

All of these criteria (with the possible exception of matching tone, a lighting issue) have manifested themselves in the work I am presenting here. They have helped to guide and analyze the cuts that are used throughout *Dogmatic*.

Sensory information & staging

When the user has complete control of the direction of view, the problem is assuring that the conditions are right to create a smooth cut. It makes no sense for example, to cut to a close-up of an object that is not currently in the frame. The primary way this is done is to monitor the state of the world and the user waiting for the correct conditions. To do this a number of different types of sensors are used. These sensors provide information like the following:

- Is "A" visible to the user.
- Where in the frame is "A"
- How long has "A" been visible.
- How close is the camera to "A"
- How close is "A" to "B"
- How long has it been since the last cut.

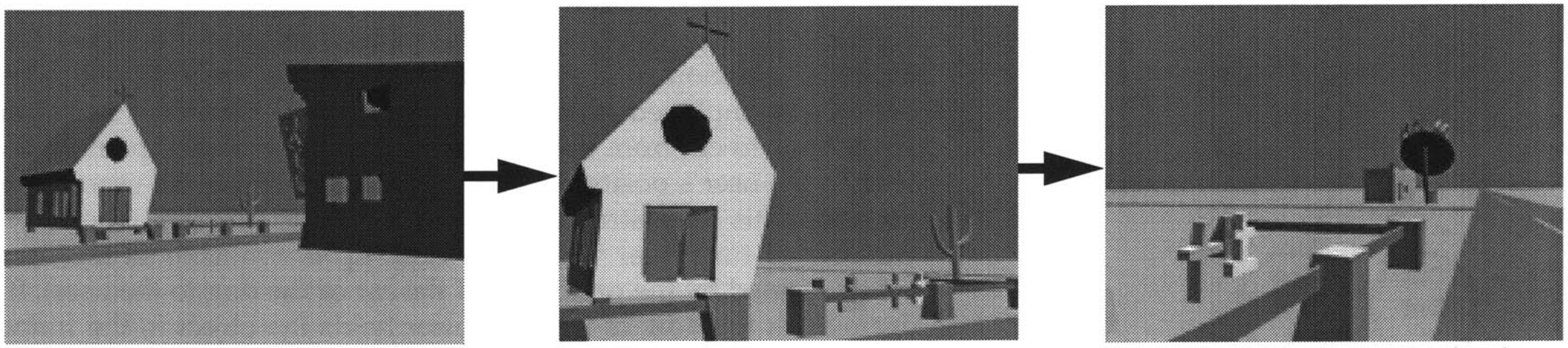
It is not always enough to just monitor the user waiting for an opportune moment. At times it becomes necessary to alter the staging of an event to help create the right moment. A simple example of this is the use of spatially located sound. A sound might be played in the user's left ear urging him/her to turn left to see the source of the sound. Another example is to move the action, such as instructing the dog to walk in front of the camera.

Examples

Armed with these tools I now want to present several examples drawn directly from *Dogmatic* and explain how they were executed.

Establishing shot

In addition to steering the user's attention, editing in film is often used to construct the story space. A shot or sequence of shots can quickly and naturally inform the audience of the spatial layout of the environment. They will build a mental map of the objects and their spatial relationships. It is because of this map that they can better understand later cuts. A traditional film method for accomplishing this is the establishing shot. The camera in this shot, near the beginning of the scene, is positioned at a distance from the objects and characters. It allows the audience to readily see all the spatial relationships. The notion of an establishing shot is used in a number of places in *Dogmatic*. In particular, the whole opening/title sequence at the beginning serves to establish the setting for the rest

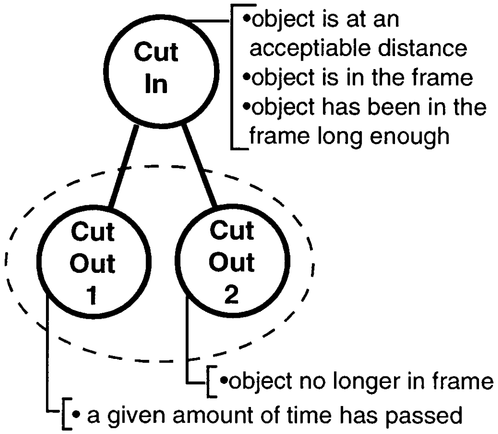


The frames from the opening establishing shot that also serves as the title sequence. The user is walked through the setting.

of the piece.

This opening is much like a long traveling shot. But due to the interface it has the added ability to inform the user of his/her position. The interface allows the users to control their direction of view which allows them to gain an even richer sense of their spatial relationship to the environment. This same notion of positioning the users at a distance, and then giving them time to look around before the action starts up again, is used as the beginning of most scenes.

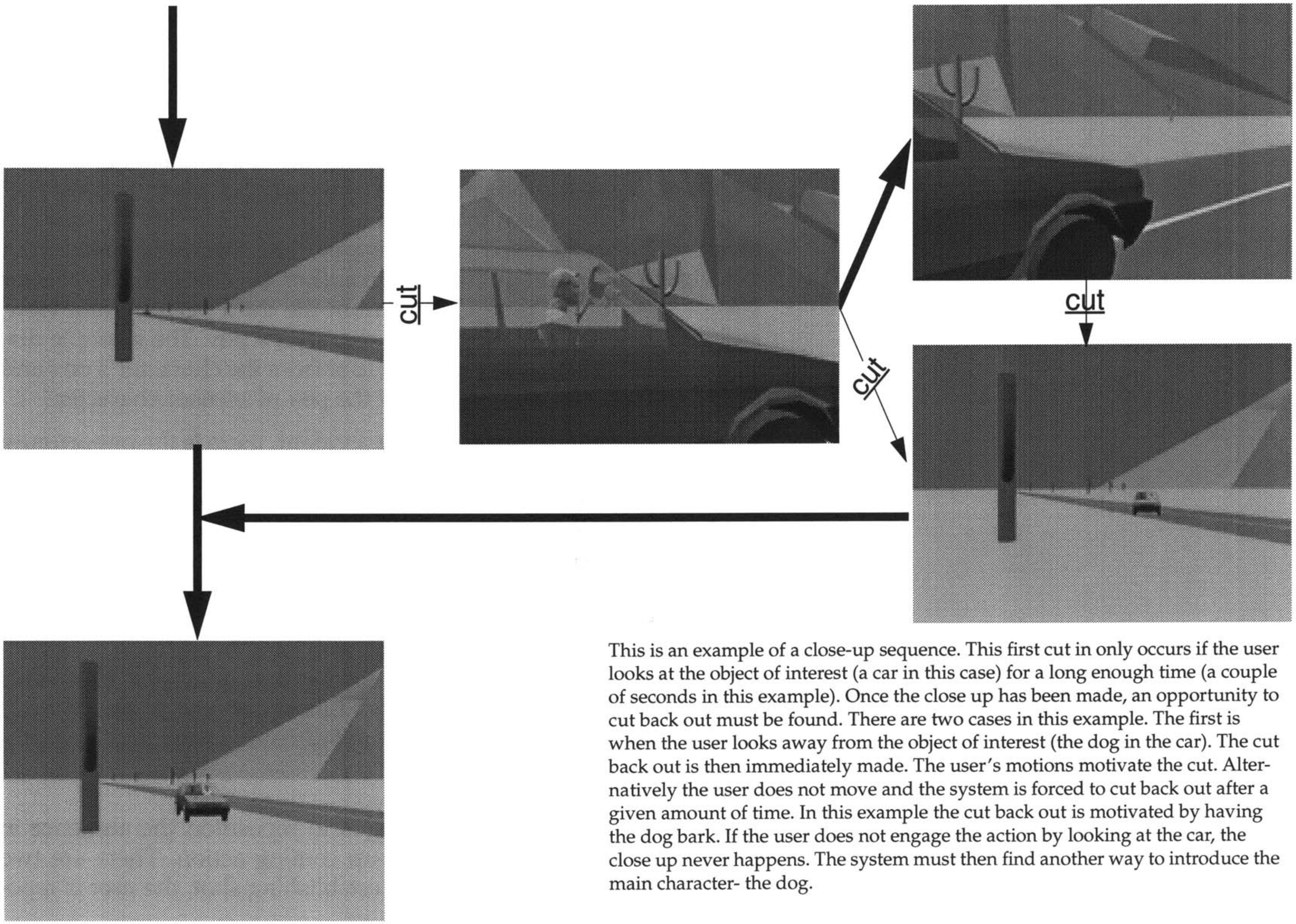
Close-up



This is the basic structure for all close-ups in *Dogmatic*. The conditional reasons for selection are list for each of the three events. Note that the two cut out events are mutually exclusive.

The close-up is probably the most common method for bringing the viewer's attention to a particular object, character, or event. The close-up is employed several times in *Dogmatic*. The most common situation is to cut from the user's position to the close-up and then back to the user's position (sometimes repositioning the user in the process). This basic structure is outlined in the diagram. There is a window of time over which the close-up is available to the user (often defined by the proximity of the car or the dog to the user). If, during this window of time, the user holds the object in the frame for a sufficient amount of time the close-up cut is then made. Once in close-up the system must decide when to cut back out. Since the user still has control over the direction of view it is possible for the user to look away from the object in close-up. This is a natural time to cut back out. The alternative is to cut back out after a predetermined amount of time. I have found that if the time based exit cut is used, it is most effective when motivated by a sound cue.

The example diagramed above is from the beginning of *Dogmatic*. The car approaches from the distance and then honks to get the attention of the user. The close-up occurs if the user looks at the car. This close-up serves to both introduce the dog and to establish its



This is an example of a close-up sequence. This first cut in only occurs if the user looks at the object of interest (a car in this case) for a long enough time (a couple of seconds in this example). Once the close up has been made, an opportunity to cut back out must be found. There are two cases in this example. The first is when the user looks away from the object of interest (the dog in the car). The cut back out is then immediately made. The user's motions motivate the cut. Alternatively the user does not move and the system is forced to cut back out after a given amount of time. In this example the cut back out is motivated by having the dog bark. If the user does not engage the action by looking at the car, the close up never happens. The system must then find another way to introduce the main character- the dog.

relationship to the person in the car. If the user chooses to not look at the car then the close-up is not triggered and the plot must find another way to do this.

Point of view shot

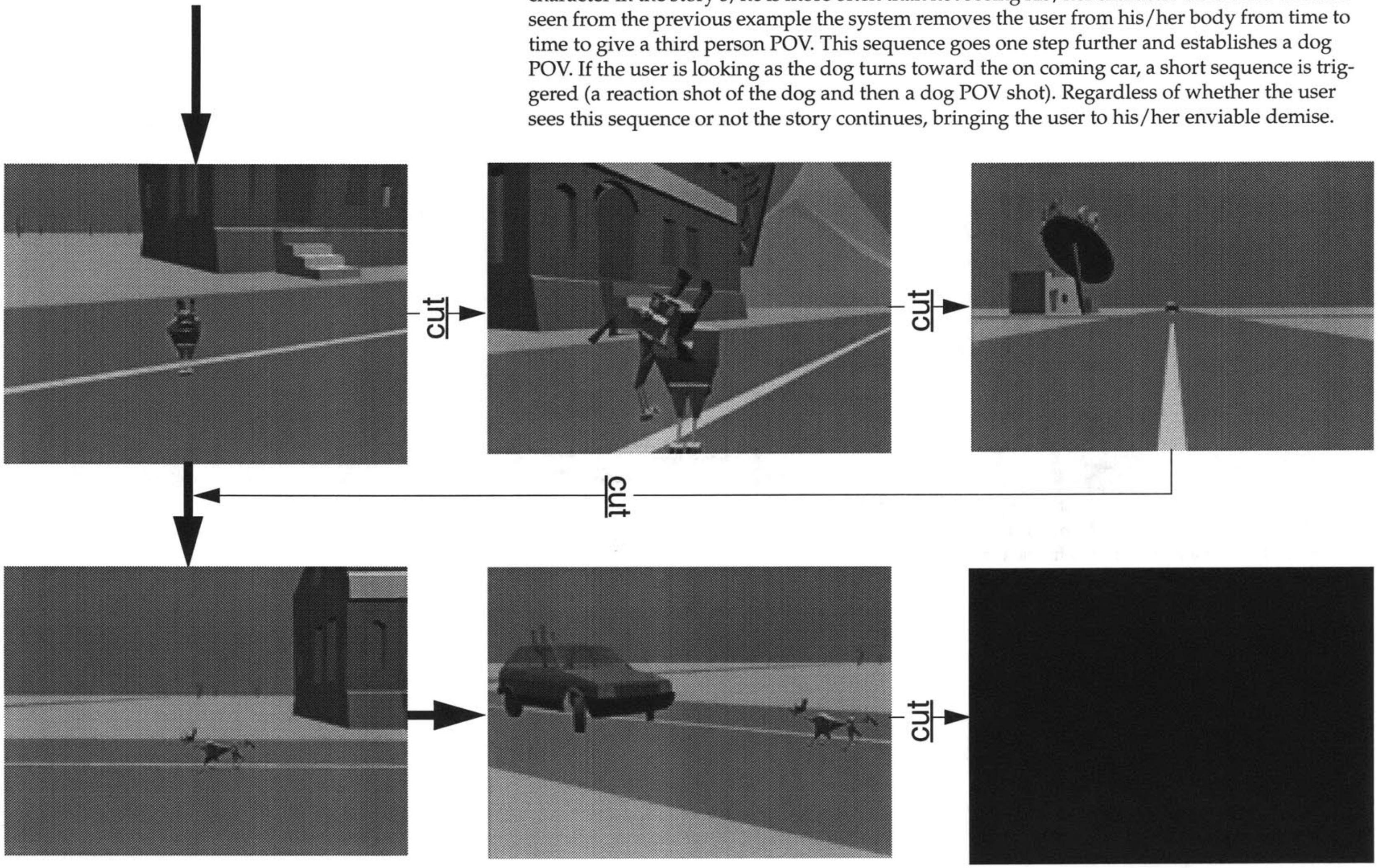
Throughout *Dogmatic* the user is in the role of a character and therefore most of what s/he sees is from the point of view (pov) of that character. We have seen from the close-up example that the user can temporarily be removed from this pov. The user is removed from this virtual body and guided by the author. In most films the viewer spends most of the time looking from the third person point view, and only occasionally moves to a character's pov. The example discussed here is about removing the user from the character's pov and then momentarily giving him/her the pov of another character.

In the moment just prior to the car accident, there is the opportunity for the user to trigger a sequence which provides a dog's point of view shot. If the user is looking at the dog as it turns in response to the honking car, the system cuts to a close-up reaction shot. Then after only a second or two the camera reverses angle and cuts to a dog's point of view shot, as the on coming car approaches. This technique of showing a reaction shot to setup a point of view shot is common in film and seems to work well here. In addition, user control of the camera is removed during this brief pov shot. This strengthens the understanding that this is not just an out of body shot, but is a shot under the control of different character.

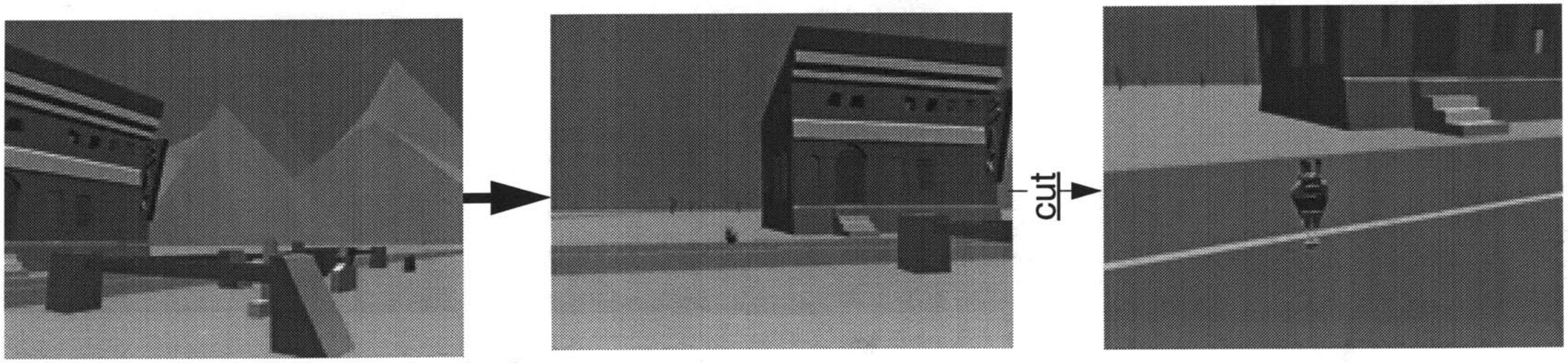
Repositioning the user

A common motivation for cutting is to reposition the audience in order for them to better see the up coming action. There are two cases in *Dogmatic* where, after an establishing shot, the user is repo-

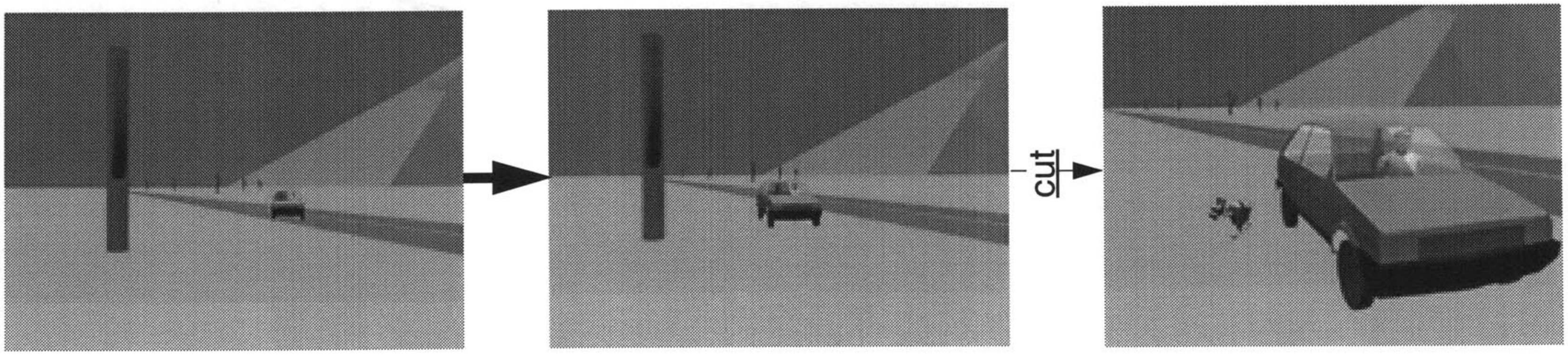
This example shows how a point of view (POV) shot can be established. Because the user is a character in the story s/he is more often than not seeing his/her character's POV. As we have seen from the previous example the system removes the user from his/her body from time to time to give a third person POV. This sequence goes one step further and establishes a dog POV. If the user is looking as the dog turns toward the on coming car, a short sequence is triggered (a reaction shot of the dog and then a dog POV shot). Regardless of whether the user sees this sequence or not the story continues, bringing the user to his/her enviable demise.



sitioned to be closer to the impending action. These cuts, like the close-up cuts, will not happen until the user is looking in the right direction. In the first example the user turns her head to bring the dog into the frame. Once the dog has been in the frame for a couple seconds, the cut is made. This cut has only two sensors checking for visibility of the dog and monitoring how long the dog has been visi-



Here are two examples of using cuts to reposition the user. To bring him/her closer to the impending action. When the object of interest is held in the frame for a sufficient amount of time the cut is made.



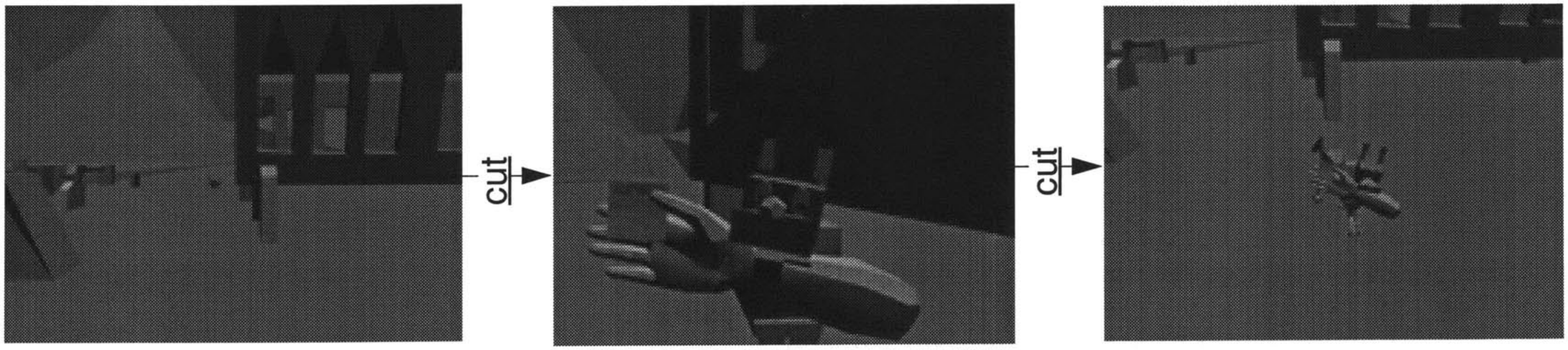
ble. The only trick is to get the user to look at the dog. In this example the dog barks at the user until s/he looks. The spatially located sound and the lack of any other action in the environment brings attention to the dog.

The second example comes right after the car has pulled over and the dog is ready to get out of the car. It cuts from a distant shot to a medium shot of the dog beside the car. In addition to repositioning the user closer so as to see the dog better, this cut also foreshortens time. This cut also over comes the need to showing the dog getting out of the car, which would have been technically difficult.

Some of the dissolves that are used as transitions between scenes are also examples of shots that reposition the user. Although not used in *Dogmatic,* the close-up is a time when the user could be repositioned. All it would take is returning the camera to the user's new position on the cut out.

Foreshortening time

The ability to lengthen or shorten the time an event takes is an important tool for controlling the presentation. Unimportant intervals can be abridged, allowing the flow of the presentation to be altered. This foreshortening of time is a powerful tool that is used a number of times in *Dogmatic.* This is illustrated by another close-up sequence like the one presented earlier but during the second cut time is foreshortened. The dog approaches and the a close-up to the dog with the arm in its mouth is made. Upon the cut out the dog is now closer to the user. This allows the action to continue without having to wait for the dog to approach. The foreshortening is important to prevent the user from getting bored. In this example the close-up is only made if the user is watching the dog. If the user is not looking at the dog we do not want to foreshorten the time



This close-up sequence shows how time can be foreshortened across a cut. During the second cut the user is moved back to the same position s/he was in and the dog is moved forward.

because the user needs more time to find the dog and engage the story.

Of course the most radical foreshortening of time in *Dogmatic* is the cross dissolve. This is a common technique, showing the passage of a long amount of time (hours in this case).

Pacing

Cuts are used to manipulate time. By manipulating time the pacing of the presentation can be adjusted. In addition, merely increasing the frequency of the cutting is a way to increase the energy of the presentation and this generates an increase in the pacing. These techniques are used in *Dogmatic*. The second scene (when the car and dog approach) and the fifth scene (when the car hits the user) both have a dramatic increase in the number of cuts. It is these scenes where the flow of the presentation is increased.

Taking control way from the user

Throughout *Dogmatic* the user is given complete control of the direction of view of the camera until the end. When the user is hit by the car the system cuts to black and control is taken from the user. As it fades back in, the user is given back limited control. They are now only allowed to look within a limited space to the left, right, up, and down. However, limiting the user's control does help assure the user will see the action during this closing shot. It is motivated by the story content. After you have been hit by a car and lay on the ground, it is only natural that you would have a limited range of movement.

Summary

This work is the first to introduce cutting into immersive environments. As these examples show we can use guidelines from the cinematic tradition to help us make cuts that maintain continuity. It is also clear that sensors and staging can be employed to account for the fact that in the immersive environment the user has at least some control of the camera. While this work is by no means a complete exploration of editing while immersed, it does provide a proof of concept and a framework for continuing this research.

Chapter Eleven

Directable Characters

A computational representation of the physical elements is needed in a virtual environment. These elements are used to create (render) the images that are seen by the audience. The field of computer graphics has developed any number of methods for representing and animating geometry. But only recently has that research begun to unify these elements to create self-contained representations of characters. These are unified models that encode the geometric pieces, the motion elements, and a behavioral representation of the characters. When these complete models of character are created, the characters can then be used much like a actor is used in a production They can be directed to perform as needed for the sake of the presentation. A directable character is endowed with many abilities and can both take direction from an outside force and perform reasonable actions autonomously when not directed.

This chapter introduces the concept of directable characters and discusses their use in the context of narrative guidance. The architecture developed to represent these characters was joint work done with Bruce Blumberg and is described in greater detail in the co-authored appendix.

Autonomy & directability

An autonomous character has both an awareness of its environment and a model of its internal state. Together these qualities allow the character to be self motivated. By looking to their current motivations and assessing the environment they can construct both goals and ways to achieve those goals. For example if the dog is hungry it will actively sense its environment in an effort to find some food. Upon smelling the food it will employ the ability to walk over to the food while also avoiding any intervening objects. Having a model that provides a character with a sense of autonomy is desirable. It means the details of the character's actions do not have to be attended to. The dog knows he is hungry and he will do the "right" thing. While autonomy abstracts the details of a character's behaviors, removing the burden of detailing every movement, pure autonomy may remove too much control from the director.

The director of a narrative guidance system wants to be able to instruct its characters at whatever level is appropriate for the given situation. At times it is enough for a character to be told it is happy. The character is the free to take what ever action it needs to express this happiness. The director at this point may not care what action it takes as long as the goal of expressing happiness is satisfied. While at another time the director may have very explicit instructions i.e. "sit down and look over there." These are commands that map more directly to the actions of the character. These examples illustrate that the characters need to be directable at a variety of different levels and the director needs to be able to seamlessly move between these different levels.

When the character provides this seamless, multi-level direction it frees the computational plot structure in this narrative guidance system to take as much or as little responsibility for the character's actions as is desired. When less care is needed high level commands can be given, and when precise control is needed low level command are used. The system can give both motivational and explicit motion directives knowing both will have the appropriate affect.

Using the characters

Characters constructed in this architecture compose a significant portion of the presentation level representation. As is detailed in the approach chapter, the presentation level is the level that represents the geometry and motion of the environments. With the exception of the music system, these pieces are encoded as character. This means that many pieces that would normally not be thought of as characters are constructed as characters in this system. In *Dogmatic* for example the lights, camera, and the user are in fact characters in addition to the obvious dog and car. This allows all changes in staging and transitions provided by the plot level to be executed as commands to characters. The camera itself is a character with limited abilities. It takes input from the user that adjusts its direction of view. It also takes direction from the computational plot to change its location and heading. These directives are how "cuts" are made.

Of all the characters used in *Dogmatic* the dog is the most sophisticated. It has a large skill set and an extensive behavior system. Therefore I will use the dog to show the different levels of direction used by the system. There are three levels:

- Motivational Level - "you need to pee"
- Task Level - "go over to the user"
- Direct Level - "wag your tail"

In the first scene the car approaches with the dog in it. Once the car pulls over, the dog is re-positioned outside of the car and is told that it "really needs to pee." This simple directive results in the dog locating a good object to mark (the cactus), going over to it, and relieving itself. Only the this high level directive is needed. When the story was authored it was decided that it was important for the dog to take this action but it was not important how he would go about it. Therefore this one motivational command was all that was needed.

If the user does not look at the dog the system will instruct the dog to approach the user. This is accomplished by giving a task level directive, "go to the user." This is a directive that tells the dog what to do but not how to do it. The dog then employs walking and turning to head toward the user's location. At this level the dog still needs a sense of its environment. It needs to find a bearing toward the user and it needs to avoid objects in its way as it heads off in that direction.

When the dog gets within an acceptable distance from the user it is instructed to stop, look at the user, and bark. This barking persists until the user looks at the dog. These commands are low level direct commands. The dog ignores motivational states that might conflict with these direct command from the plot and continues with his autonomous behavior only after a directive is given releasing him. Once the user takes note of the dog, the plot instructs the dog to go on about his business. The dog does just that as he heads off toward the cactus.

These multiple levels of directions are an important part of successful narrative guidance. A purely autonomous representation of character would not allow the plot to adjust the events to guide the user. Alternatively, without the higher level task and motivational commands, the plot would be burdened with the need to provide too much direction.

Chapter Twelve

Conclusions & Future Work

Set forth at the beginning of this thesis was the problem of bringing together highly interactive, immersive interfaces with the compelling presentation of narrative. At first the freedom of this type of interaction seemed to be at odds with the high level of structure that narration imposes. However the promise of finding a way to do this would allow a new form to emerge which contained many of the best qualities of both. This thesis has detailed an approach to this problem, as well as an application of this approach, in the story *Dogmatic*. In addition, principles or guidelines for the narrative guidance of interactivity have been presented. And a good portion of the thesis has been devoted to detailing the computational architecture developed which focused on the computational plot structure, the support of transitions, and the representation of directable characters.

In this concluding chapter I will discuss the contributions of this thesis and in that context define some areas for future work. The first two sections will discuss transitions and the semantics of a narrative guidance story such as *Dogmatic*. Last the issues behind scaling this approach to make it appropriate for both larger stories and new story forms will be explored.

Transitions

This work has shown that transitions in immersive environments can be an effective way to manipulate the user and direct his/her attention. Transitions in traditional film are used to bring the authorial voice to the presentation. Likewise, when used in immersive environments, they can be used to guide the user's attention. Even more importantly this work has shown that these transitions (changes in the visual field) can be done seamlessly when used in conjunction with music and sound staging. This technique offers the ability to manipulate the time and space of the story presentation and allows the story system to guide an audience through the narrative.

The aesthetic of the transitions demonstrated in *Dogmatic* borrow from the knowledge and techniques refined over decades of use in traditional film. But we know more can be done within the virtual environment. The technology allows us to do things that could never be done before. Now that this work has established that manipulations of time and space can be used effectively, and has built an architecture to support the development of these techniques, we can begin to explore the creation of new techniques that may be entirely unique to this medium. These new techniques will develop hand-in- hand with new conventions and rely on the evolution of the audience as well as the art form.

Semantics of story

There are a number of different approaches to interactive narrative. This work has explored one. Because the field is so young there is room for new story forms to be developed and with these forms new semantics. The conventions are still being developed. Only after these new forms mature will we be able to look back at them and analyze these semantics with any completeness. But in an effort to position my work I will briefly try to outline those I have come to understand. By doing this I can position *Dogmatic* as well as propose some future forms.

Reliability of story

Whenever an interactive story is made, it must in some way be adjustable. That is to say that the story must, at some level, change the content and/or the presentation in reaction to the user's input. It is these changes that effect the reliability of the story. In traditional story forms like cinema there is only one presentation and one story thread. The same story is told every time. It is completely reliable. In film this reliability can be attributed to production, performance, and editing. Once these elements are set, the reliability, for bad or good, remains fixed. With the introduction of interactivity the reliability of the story changes.

A story has structure at a multitude of levels. I have articulated two levels: the plot level and the presentation level. Reliability is an issue at each of these levels. In other words, if only the presentation of the narrative changes, the plot can remain fixed. In this approach the low levels of the narration are not reliable and will change with each viewing, while the core plot structure will remain unaltered with each visit. Alternatively, there are multi-threaded interactive stories.

A simple example of this is a story whose ending changes based on the actions of the user. The author has the burden of deciding at what level and in what ways a story will or will not be reliable. With each decision, they are sculpting the semantics of their story.

With an immersive interface this notion of reliability goes hand-in-hand with the amount of control the system exerts. In *Dogmatic* only one story is told but the elements of the presentation are free to change either directly or indirectly in response to the user's actions. For example, the user is given complete control over his/her direction of view. The user manipulates the camera's degrees of freedom dictating its orientation. There are many degrees of freedom in the story space that the user does not have control over: the actions of the dog and the car, the lighting, and even the camera's position. These degrees of freedom are manipulated by the system to assure that the desired story is told. The system manipulates many of the elements in the story to assure the plot points are fulfilled. This assures the reliability of the story at the plot level. But what would happen if we gave the user even more control? These degrees of freedom are resources and as more are given to the audience less are available to the system. This means the system has fewer resources at its disposal to ensure reliability. This suggests a trade off between a high degree of interactive freedom and the reliability of story.

The freedom of interaction has two faces. These are the actual amount of freedom that a user has as opposed to the perceived amount of freedom. While it is true that a greater amount of freedom given to a user does trade off with the amount of narrative structure, it is also true that a user may perceive they have a great deal of freedom when in fact they do not. Take for example a story where the user is free to look where ever s/he chooses. The system can leave them with this freedom and let them look around until they stumble upon an important story event. Alternatively, the system could orchestrate the characters and events to assure a story goal is met. This is done in *Dogmatic*. The goal of the first scene is to introduce the dog to the user. If the user engages the dog and watches it this goal is fulfilled and the story moves forward. If the user refuses to look at the dog, the dog approaches the user and barks to get his/her attention. Likewise, if the user still does not look at the dog, the dog could then take more aggressive action. The user perceives they have the freedom to look anywhere, but the system is going to ensure that they see what they need to see.

User's relationship to story

The user's relationship to the story is an important part of the semantics of a story. With immersive interfaces we have an unprecedented ability to embody the user as a character in the story. But the term "first person" is not sufficient to explain this relationship. There are two parts to this relationship. First is the issue of whether or not to position the user as a character in the experience. Do they have a presence, an embodiment? This is the relationship between the user's "I" and the story. Second is the relationship between the user's "eye" and the camera, through which the experience is seen. It is possible to establish the user as a character while his/her view of the world is from a different perspective. This is common in video games. Alternately, the user can see the world through the eyes of the character in which they are embodied. There are a number of games that do this at least part of the time. In *Dogmatic* I have chosen to embody the user as a character. Most of the time they are looking through their own eyes as they stand within their own geometric incarnation. But from time to time the system, speaking with an authorial voice, momentarily removes the users from their bodies allowing them to see with a different eye.

In *Dogmatic* it is through the "act of interacting" that the user feels embodied. And it is through the story that the risk and consequences of interacting are established. Until the end the user's role is relatively passive. You, as the user, stand as an observer occasionally approached by the dog. But at the end you are hit by the car, something that could only happen to you because of your embodiment. When the car reaches you the screen cuts to black and your ability to control the camera is removed. Next the screen fades in and you find yourself sideways on the ground with only partial control over the camera. Throughout the story you learn to trust that you have control over the camera. Even when the system removes that control for brief moments it is always restored. In this closing sequence that trust is violated. But this is necessary for the story to remain true to your embodiment. You lose control because you are the "I" in the story. For the first time the story is empowered with the ability to reach out and "bite" you, the person interacting; the audience. This suggests a new relationship between story and viewer. New techniques and conventions that speak directly to the user can be developed. There is no doubt in my mind that *Dogmatic* has only touched on this potential and it warrants more attention.

Throughout any story there are pivotal events which are necessary to the telling of the story. In addition, there are ancillary events which establish the environment and the setting of the presentation. When the audience is bestowed with the ability to interact, allowing them to change or adjust the events presented, will they be given the ability to change essential or ancillary events? One approach is to allow the system to orchestrate the essential events, not allowing the viewer to alter the core of the story. This means that the interactivity is the process of allowing the user to effect only the ancillary events. The concern is that under these conditions the user feels like and outside observer and is therefore not as emotionally affected as they would be if actively engaged in the essential events. Therefore the effort in *Dogmatic* was to engage the user's interaction in the main events; but doing this without allowing the user to derail the main story thread.

Dogmatic

Although it was beyond the scope of this thesis to do extensive user testing, I would like to report some of the reactions I have received. It is hard to evaluate the aesthetic success of a piece like *Dogmatic* because it is so subjective. But I have interviewed users in an effort to better understand whether or not they found the story readable (understandable) and to get a handle on what type of emotional responses they had during the viewing. At first I found that certain scenes were not understandable by many audience members. After adjusting and testing several things, I came to the conclusion that it was a product of the staging or inappropriate transitions. Once these places had been tuned the experience seemed to be quite readable.

As for the emotional reactions there are a couple examples worth noting. For many users I made the mistake of explaining that there was an opening traveling sequence at the beginning. When I stopped doing this I had a couple of visitors report that they felt anxious and suspenseful during this time. They looked around in anticipation of the action and they were constantly concerned that they would not be looking in the right place at the right time. This, like the second case I want to discuss, is an example of an emotional response that is a product of the user's ability to interact.

The second example occurs during the night scene. This scene is very difficult, if not impossible, to understand unless you are the one interacting. (It is common when demonstrating *Dogmatic* that only one person is interacting while others watch over his/her shoulder.) The dog walks off in response to the sound of a stick cracking. Most people watch the dog as it walks away. The system then pauses, waiting for the user to look away from the dog. At the moment they look away the a dog yelps and disappears. Users then look around somewhat frantically searching for the dog. This is a place in the story where the system is forcing a particular experience on the user. The dog will not disappear until the user looks away regardless of how long they look. This sensation of having just missed an important event is imposed. These are both examples of emotional responses that could only happen as a product of the ability to interact. It is only because the user controls the direction of view of the camera that s/he can be made to feel this way. These examples are simple but effective and suggest that this new medium can indeed reach an audience in ways that traditional storytelling cannot.

Scalibility

This architecture and its computational elements have proven effective for this particular story. But there is the question of how it will scale with respect to different story structures with different semantics. While it is unreasonable to predict what the new and future story forms will be, I would like to look at three different ways in which this type of story could be scaled and examine the ability of this architecture to accommodate it.

Multi-variant stories

Dogmatic is a short (7 to 8 minutes) and relatively narrow story experience. Regardless of your actions, six scenes happen in one particular order. You meet the dog, see the note, and are killed in the same way each time. It is very reliable. Alternatively, there is room for each of the scenes to have a multitude of ways to play out while still assuring their particular goals and plot points are met, even to the extent of introducing more than one way for the user to be killed. This would add a breadth to the experience while retaining reliability. In this architecture this could be easily accomplished with a larger production effort. The author is responsible for creating the event hierarchy (the computational plot specification). While there are seldom more than two or three possible alternatives, it would be easy enough for an author to take time to produce more and fold them into the hierarchy. Likewise, if a longer story was to be told it would be necessary to author a larger event tree.

Multiple threads

In *Dogmatic* all the events are authored to assure that the same story is told. This was an aesthetic choice that I made as an author. The system itself does not impose this. As a matter of fact the burden is on the author to assure this is the case. Alternatively, there are a number of interactive story systems that allow for multiple threads. For example a common case would be for the story to have multiple endings. There is nothing to preclude this in the computational structure of this architecture. Like the previous example of adding more breadth to a story, the onus in on the author. S/he is responsible for constructing a multitude of alteratives and specifying the criteria for choosing one over the other. Once this is done the system will traverse the event structure and make interactive decisions about which thread will be chosen. Note that a multi-threaded narrative by its nature has many variations, and these variations greatly impacts its reliability.

Multiple users

The last example of scaling that I want to examine is one of having multiple users in the same story. This issue is more complicated than the previous two. It seems to me that this will require not only additions and/or changes to the architecture, but also a new story model. Does each user experience his/her own story? Does only one user become the focus while others take more passive roles? Or do all the individual users experience parallel action that is then woven back together into one larger story? It is only after questions like these are answered that the architecture can be evaluated for its ability to represent this new story type. This evaluation is the process of asking questions like: Is the existing computational plot model sufficient to regulate each individuals interactions? Can there be and is there a need for a higher level, parallel structure that can weave these individual efforts together?

Higher level abstraction

To more easily accommodate this scalibility and to make the author's job easier, there are a number of places in the architecture that would benefit from building higher levels of abstraction; in particular more explicit models of the user, more autonomous or intelligent camera representation, and development of authoring tools.

In *Dogmatic* the model of the user is quite implicit. The sensors are simplistic and they are directly used to govern the selection of events. Alternatively, a richer and larger set of sensors could be developed. For example a higher level sensor that looks at many factors and evaluates whether or not the user is attentive could be added. This would allow an author to construct story structure unburdened with these details. Likewise, repeated computational structures like the "close-up" detailed in an earlier chapter suggest the ability to abstract out higher level camera behavior. Since the camera is represented as a directable character in this architecture these abilities could be added as behaviors to the camera character, allowing it to be re-used from story to story.

At both the plot and presentation level this current implementation relies on the author to specify things in scripting languages and sometimes by writing code. While this is fine and necessary for this first system, there is plenty of room for refining these techniques and researching the development of author tools. These tools would not only allow the author to work more efficiently but would also make this authoring process more accessible to others. In particular the plot authoring and character building steps could benefit from such tools.

There is a wealth of future research topics to be explored. While *Dogmatic,* and the architecture developed in this work, is by no means exhaustive in its success in bringing immersive narrative to light, it does serve as a foundation from which future work can grow.

References

Anderson, J. J. (1989). *Lippman on Interactivity.* MacUser. 91-96.

Arijon, D. (1976). *Grammar of the Film Language.* Los Angeles, Silman-James Press.

Armstrong, W. W. and M. Green (1987). *Near-Real-Time Control of Human Figure Models.* CG&A. 7 (6): 52-61.

Badler, N., B. Barsky, et al., Ed. (1991). *Making Them Move.* San Mateo CA, Morgan Kaufmann.

Barr, A. (1981). *Faster Calculation of Superquadric Shapes.* CG&A. 1 (1): 101-122.

Barr, A., B. Currin, et al. (1992). *Smooth Interpolation of Orientations with Angular Velocity Constraints using Quaternions.* SIGGRAPH '92, 26(2): 313-320.

Bartels, R., J. Beatty, et al. (1987). *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling.* Los Angeles, Morgen Kaufmann.

Bates, J. (1992). *Virtual Reality, Art, and Entertainment.* Presence. 1 (1): 133-138.

Bates, J., A. B. Loyall, et al. (1991). *Broad Agents*. AAAI Spring Symposium on Integrated Intelligent Architectures, Stanford, CA, SIGART Bulletin. 2 (4): 38-40.

Bathe, K.-J. (1982). *Finite Element Procedures in Engineering Analysis*. New Jersey, Prentice-Hall Inc.

Blinn, J. F. (1982). *A Generalization of Algebraic Surface Drawing*. ACM TOG. 1(3) 235-256.

Blumberg, B. (1994). *Action-Selection in Hamsterdam: Lessons from Ethology.* Third International Conference on the Simulation of Adaptive Behavior, Brighton,

Bordwell, D. (1985). *Narrative in the Fiction Film*. Madison, Wisconsin, University of Wisconsin Press.

Bordwell, D. (1994). *Understanding Filmic Understanding*. University of Wisconsin-Madison.

Boulic, R., N. Magnenat-Thalmann, et al. (1990). Human Free-Walking Model for Real-Time Interactive Design of Gaits. *Computer Animation '90*. Spinger-Verlag. 61-79.

Bove, V. M. (1994). *Object-Oriented Television*. 136th SMPTE Technical Conference, Los Angeles, CA, paper 136-3.

Branigan, E. (1992). *Narrative Comprehension and Film*. London and New York, Routledge.

Brooks, F. P. (1986). *Walkthrough - A Dynamic Graphics System for Simulating Virtual Buildings*. 1986 Workshop on Interactive 3D Graphics, Chapel Hill, North Carolina, ACM SIGGRAPH. 9-21:

Brooks, F. P. (1988). *Grasping Reality Through Illusion -- Interactive Graphics Serving Science*. CHI '88, 1-11.

Brotman, L. S. and A. Netravali (1988). *Motion Interpolation by Optimal Control*. Proceedings of SIGGRAPH '88, 22(4): 309-315.

Bruner, J. (1986). Approaching the Literary and Two Modes of Thought. *Actual Minds, Possible Worlds*. Cambridge, MA, Harvard University Press.

Burch, N. (1981). Spatial and Temporal Articulations. *Theory of Film Practice*. Princeton, NJ, Princeton University Press.

Chatman, S. (1978). *Story and Discourse: Narrative Structure in Fiction and Film*. Ithaca and London, Cornell University Press.

Chatman, S. (1993). *Reading Narrative Fiction*. New York, Macmillan Publishing Company.

Chen, M., S. J. Mountford, et al. (1988). *A Study of Interactive 3D Rotation Using 2D Control Devices*. SIGGRAPH '88, Atlanta, Georgia, ACM. 22(4): 121-129.

Cohen, M. F. (1992). *Interactive Spacetime Control for Animation*. Proceedings of SIGGRAPH '92, Chicago, IL, 26(2): 293-302.

Cook, D. A. (1990). *A History of Narrative Film*. New York, Norton & Company Inc.

Coquillart, S. (1990). *Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling*. SIGGRPH '90, Dallas, Texas, (24)4: 187-196.

Davenport, G. (1981). *New Orleans in Transition, 1983-1986: The Interactive Delivery of a Cinematic Case Study.* Massachusetts Institute of Technology.

Davenport, G., T. Aguierre-Smith, et al. (1991). *Cinematic Primitives for Multimedia*. CG&A. July 67-73.

Davenport, G., R. Evens, et al. (1993). *Orchestrating Digital Micromovies*. 26 (4): 283-288.

Davenport, G. and L. Friedlander (1994). *Interactive Transformational Environments: Wheel of Life*. Massachusetts Institute of Technology.

Don, A. (1990). Narrative and the Interface. *The Art of Human Computer Interface Design*. Reading, Mass, Addison-Wesley.

Drucker, S. M. (1994). *Intelligent Camera Control for Graphical Environments*. Massachusetts Institute of Technology.

Drucker, S. M., T. A. Galyean, et al. (1992). *CINEMA: A System for Procedural Camera Movements*. 1992 Symposium on Interactive 3D Graphics, Boston, MA, ACM SIGGRAPH. Special Issue: 67-70.

Eisenstein (1949). A Dialectic Approach to Film Form and Film Language. *Essays in Film Theory: Film Form*. Harvest.

Evans, R. (1993). *LogBoy and FilterGirl*. MIT Media Laboratory.

Farin, G. (1988). *Curves and Surfaces for Computer Aided Geometric Design*. Boston, Academic Press Inc.

Ferrin, B. (1994). *Address to MAS101*. at the MIT Media Lab.

Field, S. (1984). *The Screenwriter's Workbook*. New York, Dell.

Fisher, S. S. (1990). Virtual Interface Environments. *The Art of Human Computer Interface Design*. Reading, Mass, Addison-Wesley. 423-438.

Foley, J., A. vanDam, et al. (1987). *Computer Graphics: Principles and Practice*. Reading MA, Addison-Wesley.

Galyean, T. A. and J. F. Hughes (1991). *Sculpting: An Interactive Volumetric Modeling Technique*. SIGGRAPH '91, Las Vegas, 25(4): 267-274.

Halliday, M. (1993). *Digital Cinema: An Environment for Multithreaded Stories*. Massachusetts Institute of Technology.

Hochberg, J. and V. Brooks (1978). The Perception of Motion Pictures. *Handbook of Perception*. Academic Press.

Hsu, W. M., J. F. Hughes, et al. (1992). *Direct Manipulation of Free-Form Deformations*. Proceedings of SIGGRAPH '92, Chicago, IL, 26(2): 177-184.

Hunter, L. (1993). *Lew Hunter's Screenwriting 434*. New York, Perigee.

Hutchins, E. L., J. D. Hollan, et al. (1986). Direct Manipulation Interfaces. *User Centered System Design*. Hillsdale, New Jersey, Lawrence Erlbaum Associates. 87-124.

Isenhour, J. P. (1975). *The Effects of Context and Order in Film Editing*. AVCR. 23 (1): 69-80.

Jappinen, H. J. (1979). *A Perception-Based Developmental Skill Acquisition System*. The Ohio State University, Columbus OH.

Jappinen, H. J. (1981). *Sense-Controlled Flexible Robot Behavior*. International Journal of Computer and Information Sciences. 10 (2): 105-125.

Karp, P. and S. Feiner (1990). *Issues in the Automated Generation of Animated Presentations*. Graphics Interface '90, Halifax, Nova Scotia, 39-48.

Katz, S. (1991). *Film Directing: Shot by Shot*.

Katz, S. D. (1992). *Film Directing Cinematic Motion a Workshop for Staging Scenes*. Studio City, CA, Michael Wiese Productions.

Kelso, M. T., P. Weyhrauch, et al. (1993). *Dramatic Presence*. Presence. 2 (1): 1-15.

Kochanek, D. and R. Bartels (1984). *Interpolating splines with local tension, continuity, and bias control*. SIGGRAPH '84, ACM. 18(3): 33-41.

Laurel, B. (1986). *Toward the Design of a Computer-Based Interactive Fantasy System*. Ohio State University.

Laurel, B. (1989). *A Taxonomy of Interactive Movies*. New Media News. 3 (1):

Laurel, B. (1993). *Computers as Theatre*. Reading, Massachusetts, Addison-Wesley.

Laurel, B. K. (1986). Interface as Mimesis. *User Centered System Design*. Hillsdale, New Jersey, Lawrence Erlbaum Associates. 67-85.

Lee, P., S. Wei, et al. (1990). *Strength Guided Motion*. SIGGRAPH '90, Dallas, TX, 24(4): 253-262.

Levaco, R., Ed. (1974). *Kuleshov on Film*. University of California Press.

Lorenson, W. E. and H. E. Cline (1987). *Marching Cubes: A High Resolution 3d Surface construction algorithm*. SIGGRAPH '87, 20(4): 163-169.

Loyall, A. B. and J. Bates (1993). *Real-time Control of Animated Broad Agents*. Fifteenth Annual Conference of the Cognitive Science Society, Institute of Cognitive Science, University of Colorado - Boulder, Lawrence Erlbaum Associates. 664-669.

Mackinlay, J. D., S. K. Card, et al. (1990). *Rapid Controlled Movement Through a Virtual 3D Workspace*. SIGGRAPH '90, Dallas, Texas, ACM. 24(4): 171-176.

Maes, P. (1989). *How to do the Right Thing*. Connection Science. 1 (3): 291-323.

Maes, P. (1990). *Situated Agents Can Have Goals*. Journal for Robotics and Autonomous Systems. 6 (1&2): 49-70.

McKenna, M. and D. Zeltzer (1990). *Dynamic Simulation of Autonomous Legged Locomotion*. Proceedings of SIGGRAPH '90, Dallas, TX, 24(4): 29-38.

Mohl, R. (1981). *Cognitive Space in the Interactive Movie Map: An Investigation of Spacial Learning in Virtual Environments*. Massachusetts Institute of Technology.

Monaco, J. (1981). *How to Read a Film*. New York, Oxford University Press.

Naylor, B. (1990). *SCULPT: An Interactive Solid Modeling Tool*. Graphics Interfaces, 138-148.

Negroponte, N. (1990). Hospital Corners. *The Art of Human Computer Interface Design*. Reading, Mass, Addison-Wesley.

Norman, D. A. (1990). Why Interfaces Don't Work. *The Art of Human Computer Interface Design*. Reading, Mass, Addison-Wesley. 209-219.

Owen, D. (1986). Answers First, Then Questions. *User Centered System Design*. Hillsdale, New Jersey, Lawrence Erlbaum Associates. 361-375.

Pentland, A., I. Essa, et al. (1990). *The ThingWorld Modeling System: Virtual Sculpting By Model Forces*. SIGGRAPH Symposium on Interactive 3D Graphics, 24(2): 143-144.

Pentland, A., I. Essa, et al. (1991). *The ThingWorld Modeling System.* Algorithms and Parallel VLSI Architectures, Volume B: Proceedings: 425-434.

Raibert, M. H. and J. K. Hodgins (1991). *Animation of Dynamic Legged Locomotion*. Proceedings of SIGGRAPH '91, Las Vegas, Nevada, 25(4): 349–358.

Reeves, W. T. (1981). *Inbetweening for Computer Animation Utilizing Moving Point Constraints*. SIGGRAPH '81, Dallas, Texas, 15 (3): 263-269.

Reisz, K. and G. Millar (1978). *The Technique of Film Editing*. Boston, MA, Focal Press.

Richards, R. (1992). *A Director's Method for Film and Television*. Boston, Focal Press.

Rigopulos, A. (1992). *Growing Music from Seeds: Parametric Generation and Control of Seed-Based Music for Interactive Composition and Performance*. Massachusetts Institute of Technology.

Rubin, B. (1989). *Constraint-Based Cinematic Editing*. Massachusetts Institute of Technology.

Sederberg, T. W. and S. R. Parry (1986). *Free-Form Deformation of Solid Geometric Models*. SIGGRAPH'86, Dallas, Texas, 20(4): 151-160.

Shoemake, K. (1985). *Animating Rotation with Quaternion Curves.* SIGGRAPH '85, 245-254.

Sims, K. (1984). *Locomotion of Jointed Figures over Complex Terrain.* Massachusetts Institute of Technology.

Sims, K. (1986). *Pseudo-Inverse Control of Redundant Manipulators.* MIT Media Lab.

Smith, S. and J. Bates (1989). *Towards a Theory of Narrative for Interactive Fiction.* Carnegie Mellon University.

Todorov, T. (1971). *The Two Principles of Narrative.* Diacritics. 1 (1): 37-44.

Tyrrell, T. (1993). *The Use of Hierarchies for Action Selection, In: From Animals To Animates.* Second International Conference on the Simulation of Adaptive Behavior, MIT Press.

Ware, C. and S. Osborne (1990). *Exploration and Virtual Camera Control in Virtual Three Dimensional Environments.* 1990 Symposium on Interactive 3D Graphics, Snowbird, Utah, ACM SIGGRAPH. 24(2): 175-184.

Wilhelms, J. (1987). *Dynamics for Everyone.* Fourth USENIX Computer Graphics Workshop, 49-72.

Wilhelms, J. (1987). *Using Dynamic Analysis for Realistic Animation of Articulated Bodies.* CG&A. 7 (6): 12-27.

Wilhelms, J. and B. Barsky (1985). *Using Dynamics Analysis to Animate Articulated Bodies such as Humans and Robots*. Graphics Interface '85, Montreal, Quebec, 97-115.

Witkin, A. and M. Kass (1988). *Spacetime Constraints*. Proceeding of SIGGRAPH '88, Atlanta, Georgia, 22(4): 159-168.

Wyvill, B., C. McPheeters, et al. (1986). *Animated Soft Objects*. (2): 235-242.

Wyvill, G., C. McPheeters, et al. (1986). *Data Structure for Soft Objects*. (2): 227-234.

Zeltzer, D. (1982). *Motor Control Techniques for Figure Animation*. IEEE CG&A. 2 (9): 59-9.

Zeltzer, D. (1983). *Knowledge-based Animation*. Proc. ACM SIGGRAPH/SIGART Workshop on Motion, Toronto, Canada, April 4-6, 187-192.

Zeltzer, D. (1992). *Autonomy, Interaction, and Presence*. Presence. 1 (1): 127-132.

Zeltzer, D. and M. Johnson (1991). *Motor Planning: Specifying and Controlling the Behavior of Autonomous Animated Agents*. Journal of Visualization and Computer Animation. 2 (2): 74-80.

Zeltzer, D. L. (1984). *Representation and Control of Three Dimensional Computer Animated Figures*. Ohio State University.

Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments

Bruce M. Blumberg & Tinsley A. Galyean

To appear in Proceedings of ACM SIGGRAPH'95.

Abstract

There have been a number of recent efforts to build behavior-based autonomous creatures. While competent autonomous action is highly desirable, there is an important need to integrate autonomy with "directability". In this paper we discuss the problem of building autonomous animated creatures for interactive virtual environments which are also capable of being directed at multiple levels. We present an approach to control which allows an external entity to "direct" an autonomous creature at the motivational level, the task level, and the direct motor level. We also detail a layered architecture and a general behavioral model for perception and action-selection which incorporates explicit support for multi-level direction. These ideas have been implemented and used to develop several autonomous animated creatures.

1.Introduction

Since Reynold's seminal paper in 1987, there have been a number of impressive papers on the use of behavioral models to generate computer animation. The motivation behind this work is that as the complexity of the creature's interactions with its environment and other creatures increases, there is an need to "endow" the creatures with the ability to perform autonomous activity. Such creatures are, in effect, autonomous agents with their own perceptional, behavioral, and motor systems. Typically, authors have focused on behavioral models for a specific kind of creature in a given environment, and implemented a limited set of behaviors. There are examples of locomotion [2, 5, 6, 12, 14], flocking [15], grasping [8], and lifting [2]. Tu and Terzopoulus's Fish [17] represent one of the most impressive examples of this approach to date.

Advances in behavioral animation are critically important to the development of creatures for use in interactive virtual environments. Research in autonomous robots [4, 7, 11] supports the need to couple real-time action with dynamic and unpredictable environments. Their insights only serve to strengthen the argument for autonomous animated creatures.

Pure autonomy, perhaps, should not be the ultimate goal. Imagine making an interactive virtual "Lassie" experience for children. Suppose the autonomous animated character playing Lassie did a fine job as a autonomous dog, but for whatever reason was ignoring the child. Or suppose, you wanted the child to focus on some aspect of the environment which was important to the story, but Lassie was distracting her. In both cases, you would want to be able to provide external control, in real-time, to the autonomous Lassie. For example, by increasing its "motivation to play", it would be more likely to engage in play. Alternatively, Lassie might be told to "go over to that tree and lie down" so as to be less distracting.

Thus, there is a need to understand how to build animated characters for interactive virtual environments which are not only capable of competent autonomous action but also capable of responding to external control. We call this quality "directability." This is the fundamental problem addressed in this paper.

This paper makes 3 primary contributions to the body of literature regarding animated autonomous characters. Specifically, we describe:

- An approach to control which allows an external entity to "direct" a virtual character at a number of different levels.
- A general behavioral model for perception and action-selection in autonomous animated creatures but which also supports external control.
- A layered architecture which supports extensibility, re-usability and multiple levels of direction.

An experimental toolkit which incorporates these ideas has been successfully used to build a number of creatures: a virtual dog used in an interactive virtual environment, and several creatures used in an interactive story telling environment.

The remainder of the paper is organized as follows. In section 2 we present a more detailed problem statement about building directable autonomous creatures and summarize the key contributions of our approach in addressing this problem. In section 3 we present an overview of the general architecture. In section 4 we discuss the motor system in more detail. In section 5 we discuss our approach to sensing. In section 6 we discuss the behavior system, its major components and how it supports external direction.

2.Problem statement

An autonomous agent is a software system with a set of goals which it tries to satisfy in a potentially complex and dynamic environment. It is autonomous in the sense that it has mechanisms for sensing its environment, for interacting with its environment, and for deciding what actions to take so as to best achieve its goals[11]. In the case of an autonomous animated creature, these mechanisms correspond to a set of sensors, a motor system and associated geometry, and lastly a behavior system. In our terminology, a creature is an animate object capable of goal-directed and time-varying behavior.

Deciding on the "right" action or set of actions is complicated by a number of factors. For example, due to the problems inherent in sensing and perception, a creature's perception of its world is likely to be incomplete at best, and completely erroneous at worst. There may be competing goals which work at cross-purposes (e.g. moving toward food may move the creature away from water). This can lead to dithering in which the creature oscillates among competing activities. On the other hand, an important goal may be un-obtainable, and pursuit of that goal may prevent the satisfaction of lower priority, but attainable goals. External opportunities need to be weighed against internal needs in order to provide just the right level of opportunistic behavior. Actions may be unavailable or unreliable. To successfully produce competent autonomous action over extended periods of time, the behavior system must provide solutions to these problems, as well as others.

However, as mentioned earlier, strict autonomy is not the goal. We need, in addition, to direct the creature at a number of different levels. Three levels of input, (motivational, task, and direct) are outlined in figure 1. Additionally, commands at the direct level need to be able to take three imperative forms:

- Do it, independent of the behavior system.
- Do it, if the behavior system doesn't object.
- Suggest how an action should be performed, should the behavior system wish to perform that action.

Thus, the behavior and motor systems must be designed and implemented in such a way that it is possible to support these levels and types of direction at run-time.

Building autonomous animated creatures is inherently an iterative process. This is particularly true since we are in the early phases of understanding how to build them. Ideally, a common approach should be taken for the specification of geometry through to behavior so that a developer need only learn a single framework. Lastly, an embedded interpreter is required to facilitate testing, as well as run-time direction.

2.1 Multiple levels of control

We provide an approach to control which allows an external entity to "direct" a autonomous animated creature at a number of different levels. These levels are detailed in Figure 1. By providing the ability to "direct" the creature at multiple levels the animator or developer can choose the appropriate level of control for a given situation.

2.2 A general behavior model

We propose a distributed behavioral model, inspired by work in

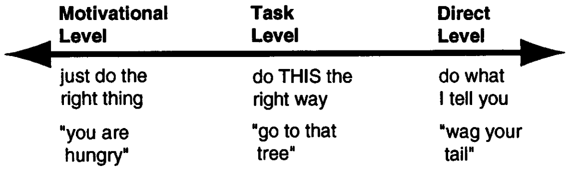


Figure 1: Here we articulate three levels at which a creature can be directed. At the highest level the creature would be influenced by changing its current motivation and relying on it to react to this change. If you tell it to be hungry it will go off looking for food. At the task level you give it a high level directive and you expect it to carry out this command in a reasonable manner (for example walking around a building instead of through it.) At the lowest level you want to give a creature a command that directly changes its geometry.

Ethology and autonomous robot research, for perception and action-selection in autonomous animated creatures but which also supports external control. The contributions of this model include:

- A general model of action-selection including an explicit model of boredom and inhibition to provide greater control over temporal patterns of behavior than previously described approaches have offered.
- A natural and general way to model the effect of external stimuli and internal motivation.
- An approach in which multiple behaviors may suggest actions to be performed and preferences for how the actions are to be executed, while still maintaining the advantages of a winner-take-all architecture.
- An implementation which supports motivational and task level direction at run-time.

We also describe a robotics inspired approach to low-level autonomous navigation in which creatures rely on a form of synthetic vision to perform navigation and obstacle avoidance.

2.3 A layered architecture

A5-layered architecture for autonomous animated creatures was created. Several important abstraction barriers are provided by the architecture:

- One between the behavior system and the motor skills, which allows certain behaviors (e.g. "move-toward") to be independent of the motor skills which perform the desired action (e.g. "drive" vs. "walk") in a given creature.

- One between the motor skills and geometry which serves as both an abstraction barrier and a resource manager.

The result is an architecture which encourages re-usability and extensibility, while providing the necessary foundation to support autonomous action with interactive direction.

3.Architecture

Figure 2 shows the basic architecture for a creature.The geometry provides the shapes and transforms that are manipulated over time for animation. The motor skills provide atomic motion elements which manipulate the geometry in order to produce coordinated motion. "Walking" or "Wagging the tail" are examples of motor skills. Motor skills manipulate the geometry with no knowledge of the environment or state of a creature, other than that needed to execute the skill. At the top rests the behavior system of a creature. This element is responsible for deciding what to do, given its goals and sensory input and triggering the correct motor skills to achieve the current task or goal. In addition to these three parts, there are two layers of insulation, the controller and the degrees of freedom (DOFs), which are important to making this architecture generalizable and extensible.

The degrees of freedom (DOFs) sit above the geometry, and can be thought of as knobs which modify particular aspects of the underlying geometry. This layer serves as both an abstraction barrier and a resource manager. In its role as resource manager, this layer insures that only one motor skill can have access to a given DOF at a time. In its role as an abstraction barrier, a DOF can correspond to a single joint, or to an articulated body where the position and orientation of the end-effector represents the DOF being modified. DOFs typically

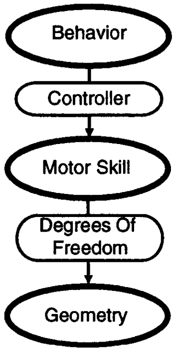


Figure 2: Block diagram of a creature's architecture. The basic structure consists of the three basic parts (Geometry, Motor Skills and Behavior) with two layers of abstraction between these parts. (the controller, and the degrees of freedom, DOFs.) The geometry is the geometric specification and transformation matrices which are used by the rendering system to display the creature. The motor skills for a creature manipulate the geometry over time via the DOFs. At the top level the behavior system issues commands to the controller. The behavior system not only allows the creature to act autonomously, but also provides a task level interface to the creature.

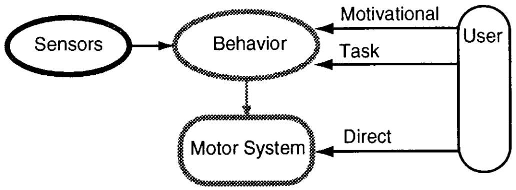


Figure 3: There are two sources of input to a creature. First are sensors associated with the creature. These sensors are used by the behavior system to enable both task level and autonomous behavior. The other source of input is from the user (or application using the creature.) This input can happen at multiple levels, ranging from simply adjusting a creature's current motivational state to directly turning a motor skill on or off.

incorporate interpolators which facilitate the task of motor skills.

Behaviors implement high level capabilities such as, "find food and eat", or "sit down and shake", as well as low level capabilities such as "move to" or "avoid obstacle" by issuing the appropriate motor commands (i.e "forward", "left", "sit", etc.) to the controller. Some behaviors may be implemented in a creature-independent way. For example, the same "move to" behavior may be applicable to any creature with basic locomotive skills (e.g. forward, left, right,...) although each may use different motor skills to perform the required action. It is the controller which provides this common interface to the motor skills by mapping a generic command ("forward") into the correct motor skill(s) and parameters for a given creature. In this way, the same behavior may be used by more than one type of creature.

Figure 3 shows the sources of input to the creature. Sensors are elements of a creature which the creature uses to interrogate the environment for relevant information. The creature may also take additional input from the user or the application using the creature. These directives can enter the creature's computational model at several different levels. At the highest level, motivational variables used by the behavior system(e.g. happiness, fear or hunger) may be specified. Task level commands may also be specified to the behavior system, for example "walk to the tree". The behavior system is then responsible for fulfilling the goal. At the lowest level a user might provide direct input to the creature's motor system (e.g. "wag tail", "walk forward", or "nod head").

4.Motor system

We use the term "motor system" to refer to all the layers that lie between the behavior system and the geometry. These parts include the motor skills in the center, and the abstraction and interface barriers on either side of the motor skills. Together these three layers of the architecture provide the mapping from motor commands to changes in the geometry over time.

4.1 Degrees of freedom (DOFs)

Each creature has a set of DOFs which are used to manipulate its geometry. The DOFs can be thought of as knobs that reposition or reshape some part of the creature. For example, a DOF might wag the tail, move a joint, or reposition an entire leg. These examples show that a DOF can be something simple (such as a single rotation) or a complex action (such as moving a foot to given location.) To do this, DOFs must function both as an abstraction barrier as well as a resource management system.Two knobs that move a front leg, one for enabling walking, and the other shaking the paw must compete for the same resources.

The resource management system is a simple one. Each motor skill specifies the DOFs which it needs in order to perform its function. A motor skill can only become active if all of its DOFs are available. If so, the motor skill's DOFs are locked. Locking the DOFs restricts it from being used by any other motor skill until it is released. A motor skill releases all of its DOFs upon completion.

Figure 4 shows two examples of how DOFs can be used to provide an abstraction barrier. Take for example, the DOF that moves a leg along a walk cycle. This knob abstracts away the need for the Motor Skill to specify how each joint should be adjusted. There are two levels of abstraction in this example. First, is the mapping from end-effector location and orientation to the positioning of the entire leg. In our work this is accomplished by using inverse kinematics (we

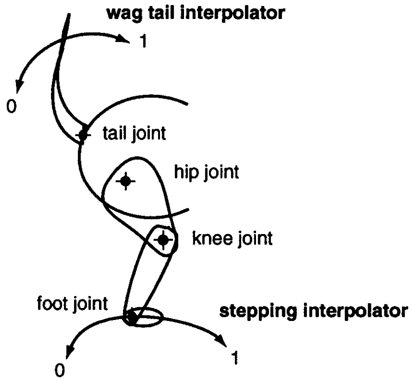


Figure 4: DOFs in a creature can provide interfaces to the geometry at several different levels. For example, joints (and therefore the associating transformation) can be directly controlled or indirectly as is the case with this leg. Here inverse kinematics is used to move the foot. An addition level of abstraction can be added by using interpolators. The interpolator on the leg provides a "one knob" interface for the motor skill. By giving a number between 0 and 1 the motor skill can set the location of the leg along one stepping cycle. Likewise the tail can be waged with only one number.

use an approach suggested by Girard [6]). In this example, one set of DOFs, the end-effector DOFs, manipulates another set, the joints. In this way, DOFs can be specified hierarchically and one DOF can lock out a series of another one. The second level of abstraction comes from mapping one knob to an end-effector position. This is done using an interpolator.

A DOF typically includes one or more interpolators which re-map a number (typically between 0 and 1) to another parameter range. For example, the DOF which controls a leg in the Dog, contains 3 B-spline interpolators: one for the trajectory of the foot, one for the joint constraints over that trajectory, and one which specifies, in effect, the relative importance of those constraints. With the interpolator installed only one number needs to be given to position the leg and any point along the walk cycle. But for the same leg to perform another action (like shaking) a new set of interpolator values must be installed. A motor skill is free to install the necessary information and interpolators into a DOF.

4.2 Motor skills

A motor skill utilizes one or more degrees of freedom to produce coordinated movement. Walking, turning, or looking-at-a-point are all examples of motor skills. A motor skill can produce complicated motion and insure that competing motor skills cannot be active at the same time. However, motor skills present an extremely simple interface to upper layers of the architecture; they can either request that a motor skill be turned on, or request that it be turned off. In either case, arguments may be passed as part of the request. The arguments passed as part of the request to turn on typically specify a goal position in some parameter space, and a speed (either in terms of time or update cycles).

Motor skills rely heavily on degrees of freedom to do their work. The DOFs are used by the motor skill to determine if it can run when asked to turn on. If one or more of its DOFs are unavailable, because they are already in use by another skill, then the motor skill will refuse to be turned on. This insures that 2 motor skills which use the same DOF will not be active at the same time. Motor Skills typically perform their task by using the interpolators provided by the DOFS. This is done in the motor skill's update method which is called automatically at the end of the creature's update cycle. Usually, the update method of a motor skill is straight forward. It may do nothing more than set a DOFs interpolation value and then check to see if it should turn off.

Turning on a motor skill which is already active is typical. In this case, the arguments may be different from when the motor skill was last turned and it responds accordingly.

Most motor skills are "spring-loaded" to reduce house-keeping on the part of a behavior system which may be using them. This means that if they have not been requested to turn on during an update cycle, they begin to move their DOFS back toward some neutral position and eventually turn off.The advantage of this approach is that a behavior, which turns on skill A, need not be written to first turn off any skill which competes with A.This will happen automatically, although it should be noted that several update cycles may be required before the behavior is successful in turning on skill A. This is consistent, however, with a reactive behavior system which re-evaluates what actions it should perform during every update cycle. It should also be noted that to facilitate interactive direction, when a behavior system is not running, this spring-loaded feature can be turned off.

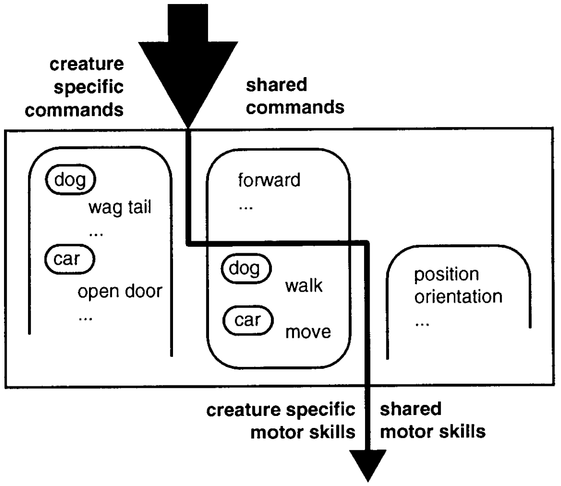


Figure 5: Each creature's controller takes in commands and maps them to the correct motor skills. There are three classes of commands. First are those that are common for many if not all creatures (on the right.) A creature inherits these commands by virtue of being a descendent of a root motor creature. Second are those that are unique to that particular creature (on the left.) These two type are generally passed thought the controller unaltered. Last, and most importantly, are those commands that are re-mapped by the controller. It is this re-mapping that provides a common interface to the creature and allows high level behaviors to be shared amongst creatures.

There are a number of basic Motor Skills which all creatures inherit, such as ones for setting the position or heading of the creature, or setting the value of a named variable etc. The basic skills are inherited by all creature from a base motor creature.

It is important to note that the design of Motor Skills is intended to be general enough that it can support a wide range of methods for generating coordinated motion.

4.3 Controller

The controller is a simple but significant layer in the architecture which serves an important function as an abstraction barrier between the behavior system and the underlying motor skills. The primary job of the controller is to map commands such as "forward", "turn", "halt", "look at" etc. into calls to turn on or turn off the appropriate motor skill(s). Thus, "forward" may result in the "walk" motor skill being turned on in the dog but the "move" motor skill in the case of the car. This is an important function because it allows the behavior system or application to use one set of commands across a potentially wide-range of creatures, and lets the motor system of each creature to interpret them differently but appropriately.

A creature's controller has a command table which contains all of the commands which the controller can be asked to execute. Each entry consists of a command name, a list of default arguments, and the motor skill(s) to turn on or off. Typically, commands fall into 3 categories: creature specific, shared, and re-mapped. These are detailed in Figure 5. In addition, the command table can contain compound commands which are a convenient way to combine commands. For example, "go straight" results in the "forward" and "stop turn" commands being executed.

Figure 6 shows how the controller accepts commands in the form of a data structure called motor command blocks. This data structure specifies the command and any arguments. The sender of the command block may request that the controller interpret the command block in one of three ways: to be executed immediately as a primary command, queued as a secondary command, or as a meta command suggesting how a command should be run. For example, a meta command might suggest that "if you are going forward, then use this gait." Secondary commands are not guaranteed to run. They are intended to be used to specify desirable but non-essential action; in other words, "do it if you have nothing better to do." A priority for a secondary command is also specified when it is issued. This priority dictates the order in which they are executed, thereby giving high priority commands the opportunity to grab the resources (DOFs) before lower priority ones. Secondary and meta-commands have proved very useful in conjunction with the Behavior System (see section 6.5)

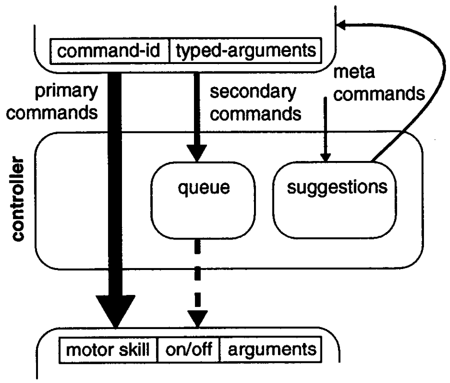


Figure 6: An incoming command is represented in a motor command block consisting of a command id and an optional list of typed arguments. If these arguments are not supplied then defaults stored in the controller are used. Any given command will turn on or off one or more motor skills, while also providing any necessary arguments. Commands take two levels of importance. Primary command are executed right away, while secondary command are queued. These queued command are executed at the end of each time cycle, and only if the necessary resources (DOFS) are available. It is expected that these secondary commands will be used for suggested but not imperative actions. The last type of input into the controller is in the form of a meta-command. These commands are stored as suggestion of how to execute a command. For example if you are going to walk I suggest that you walk slowly. These are only stored in the controller and it is the responsibility of the calling application (or user) to use or ignore a suggestion.

Commands can have return arguments as well. This allows a creature to return some information about how it will execute the commands. It also means that functions that inquire the state of a creature (its position, it velocity,...) are treated by the same mechanism as all other commands.

Secondary and meta-commands are by default ephemeral, meaning that they are only active for one update cycle and are then cleared. This requires the behavior system to continually re-assert its suggestions with each cycle, but frees the behavior system from any bookkeeping. Our implementation does support the ability to issue persistent secondary and meta commands, but this is typically only used when interaction is "direct" (without the behavior system.)

5.Sensing

An autonomous animated creature needs a sensory system, composed of one or more sensors, with which to sense its environment. Sensing in autonomous animated creatures shares some similarities with sensing in autonomous robots (e.g. potentially noisy and faulty sensors). On the other hand, autonomous animated creatures can "cheat" and directly interrogate the other creatures and objects for information. We call this direct sensing. While we utilize "direct sensing," we have also found it useful to borrow techniques from robotics and do "synthetic vision" sensing for low-level navigation and obstacle avoidance (an idea suggested earlier by Latombe[9], Reynolds[15], and others). With today's rendering hardware we have found that the use of vision techniques provides greater flexibility without a significant performance penalty.

5.1 Synthetic vision for navigation

For certain tasks such as low level navigation and obstacle avoidance, it makes sense to use synthetic vision sensors. A synthetic vision sensor renders the creature's local environment from the sensor's position and orientation. Vision techniques are then applied to the image to extract useful information.

Horswill [7] points out that while "vision" in general is a very hard problem, there are many tasks for which it is possible to use what he calls "light-weight" vision. That is, by factoring in the characteristics of the environment, the robot's interaction with the environment and by tailoring the vision task to the specific requirements of a given behavioral task, one can often radically simplify the problem. That is exactly what we are doing here.

This type of sensor can take advantage of the rendering hardware.

In addition, vision techniques developed for autonomous robots tend to be computationally cheap, easy to implement, and reasonably robust. Finally, this approach does not rely on other creatures or aspects of the environment to respond to particular queries.

Our approach is simple. The scene is rendered from the creature's eye view and the resulting image is used to generate a potential field from the creature's perspective (this is done in an approach similar to that of Horswill). Subsequently, a gradient field is calculated, and this is used to derive a bearing away from areas of high potential. Following Arkin [1], some behaviors within the Behavior System represent their pattern of activity as potential fields as well (for example, move to). These potential fields may be combined to arrive at compromise trajectories.

This sensor is a simple example of using a technique borrowed from robotics to address a specific problem. We have found that it works well in practice, is fast, is simple to implement, and is general enough to allow the our virtual dog to wander around in new environments without modification.

6.Behavior system

The purpose of the Behavior System is, at every time-step to send the "right" set of control signals to the motor system. It must weigh the potentially competing goals or needs of the creature, assess the state of its environment, and choose the set of actions which makes the "most sense" at that instant in time.

Action-selection has been a topic of some interest among Ethologists and Computer Scientists alike, and a number of algorithms have been proposed by ethologists as well as computer scientists [3, 4, 11,

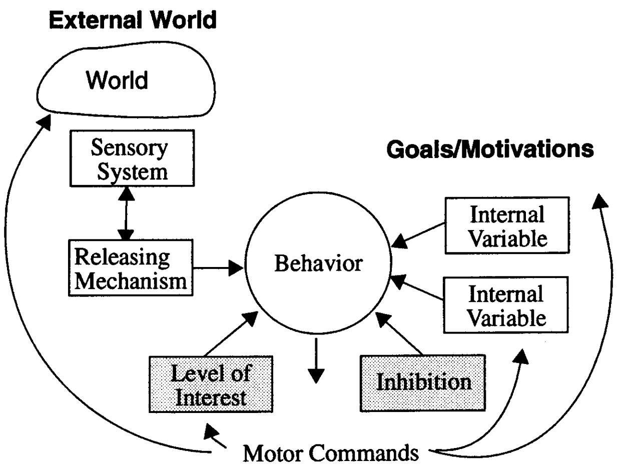


Figure 7:The purpose of a Behavior is to evaluate the appropriateness of the behavior, given external stimulus and internal motivations, and if appropriate issue motor commands. Releasing Mechanisms act as filters or detectors which identify significant objects or events from sensory input, and which output a value which corresponds to the strength of the sensory input. Internal Motivations or goals are represented via Internal Variables which output values which represents the strength of the motivation. A behavior combines the values of the Releasing Mechanisms and Internal Variables on which it depends and that represents the value of the Behavior before Level of Interest and Inhibition from other Behaviors. Level of Interest is used to model boredom or behavior-specific fatigue. As the Level of Interest drops the value of the Behavior drops. Behaviors must compete with other behaviors for control of the creature, and do so using Inhibition (see text for details).There are a variety of explicit and implicit feedback mechanisms.

16-18].

Earlier work [3], presents a computational model of action-selection which draws heavily on ideas from Ethology. The algorithm presented below is derived from this work but incorporates a number of important new features. The interested reader may consult [3] for the ethological justification for the algorithm.

6.1 Behaviors

While we have spoken of a Behavior System as a monolithic entity, it is in fact composed of a loosely hierarchical network of "self-interested, goal-directed entities" called Behaviors, each of which is in-effect fighting for control of the creature. The granularity of a Behavior's goal may vary from very general (e.g. "reduce hunger") to very specific (e.g. "chew food").

Behaviors are distinguished from Motor Skills in two ways. First, a Behavior is goal-directed whereas a Motor Skill is not. "Walking" in our model is a Motor Skill. "Moving toward an object of interest" is a Behavior. Behaviors rely on one or more Motor Skills to perform the actual pattern of activity which will satisfy the goal. Also Motor Skill typically does not require access to the sensory system. Thus, flocking is a Behavior.

On every update cycle, a Behavior calculates it's value. A Behavior's value is a relative measure of its "need" to take control and issue motor commands. The factors that determine a Behavior's value are summarized in Figure 7. The value of a Behavior may be high because the Behavior satisfies an important need of the creature (e.g. its Internal Variables have a high value). Or it may be high because the behavior's goal is easily achievable given the Behavior's perception of the environment (e.g. its Releasing Mechanisms have a high value).

Behaviors influence the system in several ways: by issuing motor commands which change the creature' s relationship to its environment, by modifying the value of Internal Variables, by inhibiting other Behaviors, or by issuing suggestions which influence the motor commands issued by other Behaviors.

6.2 Releasing mechanisms and pronomes

The purpose of a Releasing Mechanism is two-fold. First, they process sensory input (looking for an object or event of relevance to the Behavior) and produce a value which reflects the degree to which the Releasing Mechanism found matching stimuli. Typically, the value will depend not only on whether the stimuli was found, but also on the distance to the stimuli. For example, the value of a Releasing Mechanism used by the "sit" behavior in the dog is a function of whether a person is found, whether the person is performing the right gesture and the distance to the person.

In addition to transducing a value from sensory input, a Releasing Mechanism also fills in a data structure available to the Behavior called a Pronome [13]. The Pronome acts like a pronoun in English; The use of Pronomes makes it possible for the Behavior to be written in terms of "it", where the type of "it" is defined by the Behavior's Releasing Mechanism. Thus, a "stopNearAndDo" behavior callback doesn't need to care what kind of object it is stopping near.

While Releasing Mechanisms may be looking for very different objects and events, they typically have a common structure. This is described in more detail in Figure 8. The implication of this is that it is possible to share much of the functionality across Releasing Mechanisms.

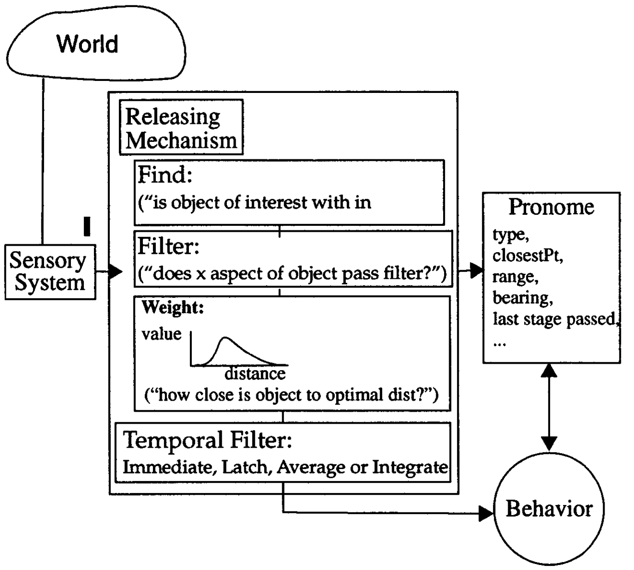


Figure 8: Releasing Mechanisms identify significant objects or events from sensory input and outputs a value which represents the strength of the stimulus. By varying the allowed maximum for a given Releasing Mechanism, a behavior can be made more or less sensitive to the presence of whatever input causes the Releasing Mechanism to have a non-zero value. A Releasing Mechanism has 4 phases (Find, Filter, Weight and Temporal Filtering), as indicated above, each of which is implemented by callbacks. Releasing Mechanisms can often share the same generic callback for a given phase. Temporal Filtering is provided to deal with potentially noisy data.

6.3 Internal variables

Internal Variables are used to model internal state. Like Releasing Mechanisms, Internal Variables express their value as a number. This value can change over time based on an autonomous growth and damping rate. In addition, Behaviors can potentially modify the value of an Internal Variable as a result of their activity.

Both Releasing Mechanisms and Internal Variables may be shared by multiple Behaviors.

6.4 Inhibition and level of interest

A Behavior's value is also affected by inhibition from competing Behaviors. We incorporate an explicit model of inhibition first proposed by Ludlow [10]. Ludlow's observation was that if the following conditions were true: behaviors are mutually inhibiting, the inhibitory gains are restricted to be greater than 1, and behavior values are restricted to being zero or greater, then once the system stabilized only one behavior would have a non-zero value.

Ludlow's model provides a robust mechanism for winner-take-all arbitration. It also provides a way of controlling the relative persistence of Behaviors. By modifying inhibitory gains, the relative level of persistence of a given Behavior may be adjusted. When the gains are low, the system tends to dither among different behaviors. When gains are high, the system shows more persistence. This is illustrated in Figure 9 below:

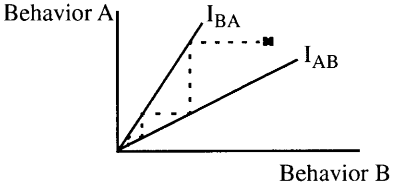


Figure 9: Inhibitory gains are used to control the persistence of Behaviors. They can be thought of as the slopes of "switching lines" in a multi-dimensional state-space. When a Behavior is active, it will stay active until its value relative to that of the other Behaviors with which it competes reaches a switching line. This is shown above in the case of 2 competing Behaviors. Behavior B is initially active, and it will stay active until its value, relative to A reaches line I_{BA}, whose slope corresponds to the Inhibitory gain B applies against A.

The use of high inhibitory gains can result in pathological behavior in which a creature pursues a single, but perhaps unavailable goal, to the detriment of less important, but achievable goals. Ludlow addressed this problem by suggesting that a level of interest be associated with every behavior. It is allowed to vary between 0 and 1 and it has a multiplicative effect on the behavior's value.When an behavior is active the level of interest decreases which in turn reduces the value of the behavior. When the behavior is no longer active, the level of interest rises. Thus, Level Of Interest provides a mechanism for implementing a form of time-sharing, in which a behavior with an intrinsically high value gives up control and behaviors with lower values are able to run.

6.5 Behavior groups

Behaviors are organized into groups of mutually inhibiting behaviors called Behavior Groups. These Behavior Groups are in turn organized in a loose hierarchical fashion as shown in Figure 10. While we find a loose hierarchical structure useful, all the Behaviors could be in a single Behavior Group. In either case, we make use of the controller's capability to accept different imperative forms to allow lower priority Behaviors to still express their preferences for actions.

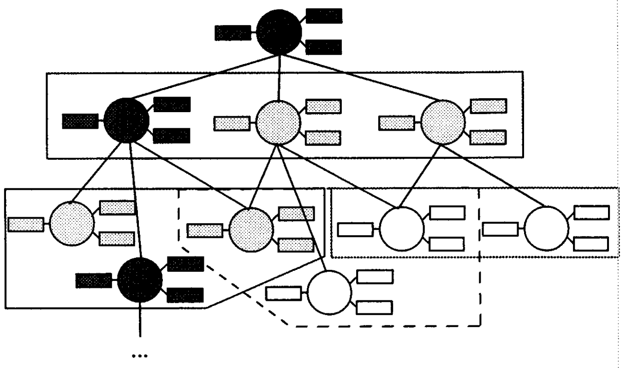


Figure 10:Behaviors are organized into groups of mutually inhibiting behaviors called Behavior Groups. These Behavior Groups are in turn organized in a loose hierarchical fashion. Behavior Groups at the upper levels of the hierarchy contain general types of behaviors (e.g. "engage-in-feeding") which are largely driven by motivational considerations, whereas lower levels contain more specific behaviors (e.g. "pounce" or "chew") which are driven more by immediate sensory input. The arbitration mechanism built into the algorithm insures that only one Behavior in a given Behavior Group will have a non-zero value after inhibition. This Behavior is then active, and may either issue primary motor commands, or activate the Behavior Group which contains its children Behaviors (e.g. "search-for-food", "sniff", "chew" might be the children behaviors of "engage-in-feeding"). The dark gray behaviors represent the path of active Behaviors on a given tick. Behaviors which lose to the primary Behavior in a given Behavior Group may nonetheless influence the resulting actions of the creature by issuing either secondary commands or meta-commands

6.6 Use of primary, secondary and meta-commands

Low priority Behaviors (i.e. those which have lost the competition for control) may still express their preferences for actions. They can do this in two ways:

- They can issue secondary commands which are queued up and executed after primary commands (those issued by the winning behavior).
- They can issue meta-commands which can be used by the winning behavior as recommendations for how to perform a given action. Gait preferences are a simple form of a meta-command. Potential Fields are another form of meta-command (see section on sensing).

For example, the Dog may have a behavior whose sole function is to alter the dog's characteristics to demonstrate to the user how the dog is feeling. In this case, the behavior may issue secondary commands for ear position, tail and mouth position, body posture, and meta-commands for gait. The reason for using secondary commands is that these are desirable actions, but not essential ones. Similarly, this behavior may not know whether the dog should go forward or not, but is in a position to offer a suggestion for how the dog should go forward.

Despite the use of secondary and meta-commands, the winning behavior still has ultimate say over what actions get performed while it is active. It can over-rule a secondary command by removing it from the queue or by executing a Motor Skill which grabs a DOF needed by a given secondary command. In the case of meta-commands, the winning behavior can choose to ignore the meta-command, in which case it has no effect.

6.7 The algorithm

The action-selection algorithm is described below. The actual equations are provided in appendix A.

On each update cycle:

(1) All Internal Variables update their value based on their previous value, growth and damping rates, and any feedback effects.

(2) Starting at the top-level Behavior Group, the Behaviors within it compete to become active. This is done as follows:

(3) Each Behavior has its Releasing Mechanisms update their value based on the current sensory input. This value is then summed with the Behavior's Internal Variables and the result is multiplied by the

Behavior's Level Of Interest. This value represents the value of the Behavior before Inhibition. This is repeated for all Behaviors in the group.

(4) For each Behavior in the group, the inhibition due to other Behaviors in the group is summed and subtracted from the Behavior's pre-inhibition value. The resulting value is clamped to 0 or greater.

(5) If after step (3) more than one Behavior has a non-zero value then step (4) is repeated until this condition is met. The Behavior with a non-zero value is the active Behavior for the group.

(6) All Behaviors in the group which are not active are given a chance to issue secondary or meta-commands. This is done by executing a suggestion callback associated with the Behavior. The callback is responsible for issuing the appropriate commands.

(7) If the active Behavior has a Behavior Group as a child (i.e. it is not a Behavior at the leaf of the tree), then that Behavior Group is made the current Behavior Group and the process is repeated starting at step (3). Otherwise, the Behavior is given a chance to issue primary motor commands.This is done by executing a leaf callback associated with the Behavior. The callback is responsible for issuing the appropriate commands.

6.8 Directing the behavior system

External direction of the Behavior System during run-time is easily accomplished due to several features of its design:

- All components of the Behavior System are subclasses of Inventor classes. Inventor supports named variables, which may then may be accessed by name. Using this

```
DEF DESIRETOPLAY InternalVariable {
    value 3.0
    growth .01 damping .001 }

...
DEF CROUCH LeafBehavior{
    leafCBName "stopNearAndDo"
    defaultCmdName “crouch”
    DependsOnVariables {
        USE DESIRETOPLAY
        DEF CROUCHRM CreatureRM {
            typeTag PERSON
            activeRange [0,  16 , 24]
            minValue 0.0 maxValue 10
        }
    }
}
```

Figure 11:The Inventor definition of part of the dog's behavior graph is shown above, in this case the definition of the Internal Variable DESIRETOPLAY and a Leaf Behavior called CROUCH. CROUCH's value depends on 2 variables: DESIRETOPLAY and a Releasing Mechanism called CROUCHRM. CROUCHRM uses the default find(), filter() and weight() callbacks to look for a creature with a typeTag "Person" within the active Range of 0-24. If a Person is found in the range, it returns a value between 0 and 10 based on the distance of the person from the optimal distance of 16. If this Behavior becomes active it will invoke the "stopNearAndDo" callback. This callback sends the "halt" command to the Dog's controller if the Dog is still moving, otherwise it sends the "crouch" command.

capability any Internal Variable may be accessed at run-time and its value changed. Thus, to increase the likelihood that the Dog will play with the user, one would simply set the value of DESIRETOPLAY to be a higher number (see Figure 11).

- Extensive use of parameterization coupled with named access makes it possible to easily change the behavioral characteristics of the creature at run-time.

7.Conclusion

Autonomy and directability are not mutually exclusive. We have detailed an architecture and a general behavioral model for perception and action-selection which can function autonomously while accepting direction at multiple levels. This multi-level direction allows a user to direct at whatever level of detail is desirable. In addition, this blend of autonomy and directability is demonstrated with several creatures with in the context of several applications in the accompanying video.

Appendix A.

Behavior Update Equation:

$$v_{it} = Max\left[\left(li_{it} \cdot Combine(\sum_k rm_{ki}, \sum_j iv_{jt}) - \sum_m n_{mi} \cdot v_{mt}\right), 0\right]$$

Where at time t for Behavior i, v_{it} is its value; li_{it} is the level of interest; rm_{kt} and iv_{jt} are the values of Releasing Mechanism k, and Internal Variable j, where k and j range over the Releasing Mechanisms and Internal Variables relevant to Behavior i; n_{mi} (n>1) is the Inhibitory Gain that Behavior m applies against Behavior i, v_{mt} is the value Behavior m where m ranges over the other Behaviors in the current

Behavior Group. *Combine()* is the function used to combine the values of the Releasing Mechanisms and Internal Variables for Behavior i (i.e addition or multiplication).

Internal Variable Update Equations:

$$iv_{it} = (iv_{i(t-1)} \cdot damp_i) + growth_i + \sum_k effects_{kit}$$

Where at time t for Internal Variable i,iv_{it} is its value; $iv_{i(t-1)}$ is its value on the previous time step; $damp_i$ and $growth_i$ are damping rates and growth rates associated with Internal Variable i; and $effects_{kit}$ are the adjustments its value due to the activity of Behavior k, where k ranges over the Behaviors which directly effect its value when active.

$$effects_{kit} = (modifyGain_{ki} \cdot v_{k(t-1)})$$

Where $effects_{kit}$ is the effect of Behavior k on Internal Variable i at time t; $modifyGain_{ki}$ is the gain used by Behavior k against Internal Variable i and $v_{k(t-1)}$ is the value of Behavior k in the preceding time step.

Level of Interest Update Equation:

$$\begin{aligned} li_{it} = {} & Clamp(((li_{i(t-1)} \cdot damp_i) + growth_i + \\ & (v_{i(t-1)} \cdot bRate_i)),0,1) \end{aligned}$$

Where li_{it} is the Level Of Interest of Behavior i at time t, and $bRate_i$ is the boredom rate for Behavior i. Clamp(x,y,z) clamps x to be between y and z. Note Level Of Interest is just a special case of an Internal Variable.

Releasing Mechanism Update Equation:

$$\begin{aligned} rm_{it} = {} & Clamp(TemporalFilter(t, rm_{i(t-1)}, \\ & Find(s_{it}, dMin_i, dMax_i) \cdot Filter(s_{it}) \cdot \\ & Weight(s_{it}, dOpt_i)), min_i, max_i) \end{aligned}$$

Where rm_{it} is the value of Releasing Mechanism i at time t; s_{it} is the relevant sensory input for i; $dMin_i$ and $dMax_i$ are minimum and maximum distances associated with it; *Find()* returns 1 or 0 if the object of interest is found within s_{it} and within $dMin_i$ to $dMax_i$; *Filter()* returns 1 or 0 if the object Of Interest matches some additional criteria; *Weight()* weights the strength of the stimulus based on some metric such as optimal distance $dOpt_i$; *TemporalFilter()* applies a filtering function (latch, average, integration, or immediate) over some period t; and *Clamp()* clamps the resulting value to the range min_i to max_i.

References

1. Arkin, R.C., *Integrating Behavioral, Perceptual, and World Knowledge in Reactive Navigation,* in *Designing Autonomous Agents,* P. Maes, Editor. 1990, MIT Press: Cambridge.
2. Badler, N.I., C. Phillips, and B.L. Webber, *Simulating Humans: Computer Graphics, Animation, and Control.* 1993, Oxford University Press.
3. Blumberg, B. *Action-Selection in Hamsterdam: Lessons from Ethology.* in *Third International Conference on the Simulation of Adaptive Behavior.* 1994. Brighton:
4. Brooks, R., *A Robust Layered Control System for a Mobile Robot.* 1986. **RA-2**.
5. Bruderlin, A. and T.W. Calvert. *Dynamic Animation of Human Walking.* in *SIGGRAPH '89.* 1989. Boston, MA: 233-242
6. Girard, M. and A. Maciejewski. *Computational Modeling for the Computer Animation of Legged Figures.* in *SIGGRAPH '85.* 1985. 263-270
7. Horswill, I. *A Simple, Cheap, and Robust Visual Navigation System.* in *Second International Conference on the Simulation of Adaptive Behavior.* 1993. MIT Press.
8. Koga, Y., *et al. Planning Motion with Intentions.* in *ACM Siggraph94.* 1994. Orlando, FL: 395-408
9. Latombe, J.C., *Robot Motion Planning.* 1991, Boston: Kluwer Academic Publishers.
10. Ludlow, A., *The Evolution and Simulation of a Decision Maker,* in *Analysis of Motivational Processes,* F.T.&.T. Halliday, Editor. 1980, Academic Press: London.
11. Maes, P., *Situated Agents Can Have Goals.* Journal for Robotics and Autonomous Systems, 1990. **6**(1&2): p. 49-70.
12. McKenna, M. and D. Zeltzer. *Dynamic Simulation of Autonomous Legged Locomotion.* in *Proceedings of SIGGRAPH '90.* 1990. Dallas, TX: 29-38
13. Minsky, M., *The Society of Mind.* 1988, New York: Simon & Schuster.
14. Raibert, M.H. and J.K. Hodgins. *Animation of Dynamic Legged Locomotion.* in *Proceedings of SIGGRAPH '91.* 1991. Las Vegas, Nevada: 349–358

15. Reynolds, C.W. *Flocks, Herds, and Schools: A Distributed Behavioral Model.* in *ACM Siggraph.* 1987. 25-34
16. Tinbergen, N., *The Study of Instinct.* 1950, Oxford: Clarendon press.
17. Tu, X. and D. Terzopoulos. *Artificial Fishes: Physics, Locomotion, Perception, Behavior.* in *ACM Siggraph94.* 1994. Orlando, FL: 45-50
18. Tyrrell, T. *The Use of Hierarchies for Action Selection, In: From Animals To Animates.* in *Second International Conference on the Simulation of Adaptive Behavior.* 1993. MIT Press.