# OBJECT-ORIENTED INTERACTIVE CINEMA

by

Adrian Jones

B.Sc., Mount Allison University, 1998

Graduate Diploma, Concordia University, 2000

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

In the Program
of
Computing Arts and Design Sciences



SIMON FRASER UNIVERSITY

April 2003

## Approval

**Name:** Adrian Jones

**Degree:** Master of Applied Science

**Title of Thesis:** Object-Oriented Interactive Cinema

**Examining Committee:**

**Chair:** **Dr. Rob Woodbury**
Professor

**Kenneth Newby**
Senior Supervisor
Assistant Professor

**Jim Bizzocchi**
Supervisor
Associate Professor

**Martin Gotfrit**
**External Examiner**
Associate Professor
Department of Contemporary Arts
Simon Fraser University

**Date approved:** April 11th 2003

## Abstract

This thesis presents a formalized conceptual model for researchers in multi-linear interactive cinema. This need for standardization is not surprising given the breadth of backgrounds of interactive cinema researchers: digital artists, filmmakers, computer scientists, mathematicians, narrative theorists, cognitive psychologists, semioticians, *etc*. This thesis has two principle objectives: (1) To provide a robust model that unifies the main concepts discussed in the emerging field of interactive cinema; and (2) To aid interactive cinema composition and multi-linear narrative generation by providing mechanisms and procedures to use as conceptual scaffolding in the construction of an interactive cinematic work. The model takes a modular approach to cinema by sub-dividing the medium into its constitutive parts, which are reassembled at run-time into a final cinematic experience based on user interaction. To achieve a strong theoretical foundation, the model is based on object-oriented modeling techniques, and thus is called the **Object-Oriented Interactive Cinema** model, or OOIC.

Since the invention and rise of new media, digital interactive storytelling has evolved considerably, but it is still far from full actualization. OOIC is a form of new media; it is a form of digital cinema; and it is part of the interactive cinema research discipline, yet it also stands apart as a unique model for incorporating interactivity into the cinematic narrative. To help lead interactive cinema into a mature art form, the OOIC model (1) Describes a systemic approach to narrative composition and structure, dividing narrative into *story* and *plot*; (2) Presents an extensible, universally applicable representational framework, based on an object-oriented class structure, to describe an interactive cinema project's media objects; (3) Presents a design methodology for incorporating graph theory, transition matrices, and state diagrams into the composition of multi-linear narratives using film objects; and (4) Establishes principles for advanced

interactivity within a cinematic context by outlining principles for user interaction and navigation.

A project that was created using the OOIC model is described at the end of the thesis. The project demonstrates the model’s applicability and effectiveness at aiding in the analysis, structure, design, and composition of a multi-linear cinematic work.

# Dedication

I dedicate this thesis to the graduate students of Simon Fraser University Surrey.

# Acknowledgements

I would like to thank the entire faculty at Simon Fraser University Surrey, whose hard work and dedication have made our emerging school something to be proud of. I would like to especially thank my advisors, Kenneth Newby and Jim Bizzocchi, for their guidance and support in the completion of this thesis. Thank you as well to my external examiner, Martin Gotfrit, and the graduate committee chair, Dr. Rob Woodbury. Special thanks to Robb Lovell for his programming and artistic skills, and his dancing feet.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1:
# Introduction

This thesis presents a formalized conceptual model for researchers in multi-linear interactive cinema. Because the field of interactive cinema is still emerging and developing – indeed, the very name is surrounded with ambiguity – researchers tend to lack a common vocabulary for discussing similar or identical concepts. This need for standardization is not surprising, however, given the breadth of backgrounds of interactive cinema researchers: digital artists, filmmakers, computer scientists, mathematicians, narrative theorists, cognitive psychologists, semioticians, *etc*.

The goal of this thesis is twofold:

1. To provide a robust model that unifies the main concepts discussed in the emerging field of interactive cinema;
2. To aid interactive cinema composition and multi-linear narrative generation by providing mechanisms and procedures to use as conceptual scaffolding in the construction of an interactive cinematic work.

True to the trend of new media, the model takes a modular approach to cinema by sub-dividing the medium into its constitutive parts, which are then reassembled into a final cinematic experience based on user interaction. To achieve a strong theoretical foundation for the modularity, the model is based on object-oriented modeling concepts. Object-oriented theories, from analysis, design, and programming, have proved, over the last few decades, to be quite effective at modeling complex systems, but so far they have, almost exclusively, only been applied to software development. The objective of this thesis is to use object-oriented modeling techniques to present a formalized model for discussing and implementing interactive cinema. I call this model **Object-Oriented Interactive Cinema**, also referred to by the acronym OOIC.

Object-Oriented Interactive Cinema (OOIC) is an attempt to create a new form of digital cinema that aims at effectively combining database and narrative. The term *object-oriented* implies a modular system, and OOIC uses the concept of an *object* comparably to object-oriented analysis (OOA) and object-oriented design (OOD) paradigms. Just as in computer programming where an algorithm specifies the steps or functions to be performed on data, the OOIC structure employs algorithms and rule-sets designed to dynamically generate navigation paths from a matrix of interconnected multimedia objects. The underlying mechanism for knowledge representation is the OOIC class structure, which specifies all of the classes, attributes, and operations used by the interactive cinema system.

In the OOIC model, an object-oriented database is composed of multimedia or 'film objects' (instances of the class **FilmObject**) that are assembled in a non-linear fashion to create a meaningful narrative. The aim is that, even though the system is using embedded film material, the resultant narrative is a pseudo-emergent property of the participant's interaction with the active cinematic environment.

### *1.1 Thesis Outline*

The Object-Oriented Interactive Cinema (OOIC) model can be seen as a unification of two pivotal concepts: object-oriented cinema and interactive cinema. The first part of the thesis discusses the terms new media and interactivity, and attempts to provide a clear understanding of their meaning in the context of interactive cinema. OOIC encompasses both theoretical and practical considerations, and as such this section not only discusses the semiotics behind the word *interactivity*, but also outlines guidelines for its implementation. The second part of the thesis (Chapter 3) offers a working definition for *narrative*, and argues that we need to re-envision the practice of storytelling when using interactive media objects. This section also discusses different types of interactivity and a framework for evaluating interactivity for a multi-linear narrative. Chapter 4 begins the discussion of digital cinema, and presents the issues that arise from the computerization of the cinema art form. Chapter 5 deals with the integration of interactivity into the digital cinema model.

Before outlining the details of the Object-Oriented Interactive Cinema model, Chapter 6 presents some of the past and current research being done in the field of interactive cinema. This discussion demonstrates how OOIC fits into the emerging discipline, and shows how it addresses particular concerns and missing elements that are needed to bring the field to a higher academic level. Chapter 7 begins discussion of the OOIC model itself, outlining basic object-oriented principles, and developing an object-oriented model for designing and implementing interactive cinema. The first section of the chapter questions how a film can be broken down and modularized into discrete and meaningful units, or objects, and discusses issues around film object granularity. The next couple of sections address multi-linear narrative and interactivity in the context of OOIC, eventually leading the discussion to a formal object-oriented methodology. As part of the object-oriented analysis (OOA) section, basic OO concepts are introduced, and a detailed class structure is presented for OOIC. The chapter ends with a discussion on temporal and spatial montage and their implementation under the OOIC model.

Chapter 8 brings up the object-oriented design (OOD) component of OOIC, and discusses how graph theory and transition matrices can be used in the construction of a multi-linear interactive narrative. Chapter 9 establishes principles for advanced interactivity within a cinematic environment by outlining principles for user interaction and navigation. This chapter also addresses issues around effective capture of user inputs, and provides the requirements for meaningful analysis and interpretation of this data.

The final section of the thesis examines an installation project that was built using the OOIC model, highlighting how OOIC was effective at aiding in the creation of a multi-linear cinematic work.

# Chapter 2:
# New Media and Interactivity

### *2.1 A New Revolution*

> *"Every age seeks out the appropriate medium in which to confront the unanswerable questions of human existence"* (Murray 2000).

We are currently in the midst of a revolution.

What exactly the revolution is, however, is still to be determined. Many people argue that we are navigating the beginning or middle of the *computer* (or *digital*) revolution. Others argue, however, that the computer revolution is simply an extension of the industrial revolution, or that – rather than focusing solely on the digital – the current revolution could more broadly be termed the *communication* revolution (encompassing both analog devises, such as the printing press, as well as digital tools). I tend to side with the theorists that advocate a revolution based on the way digital technology has impacted society and culture. It is my belief that the revolution is digital, and its effects have reached to practically all aspects of communication culture, from acquisition to distribution.

With the computer revolution comes digital communication, and therefore digital media. Digital media is often referred to as *new media*, in spite of the obvious fact that what is new today will not be new forever. Having to redefine new media every few years seems counter-productive, but nonetheless I use the term here interchangeably with digital media, as it seems to have embedded itself in society as a way of describing the media forms that are emerging from the properties of digital environments and toolsets. As Lev Manovich (2001, p.20) states, new media "represents a convergence of two separate historical trajectories: computing and media technologies."

Most media forms today have been touched by the digital revolution, some to the point of being completely converted (e.g. magnetic cassette tapes to CD's for music distribution), some on the way to being converted (e.g. analog TV signals to digital TV signals; VHS to DVD format for movie rental and distribution), and some to the point of recognizing the digital revolution but not converting to it (e.g. for live performances, most musicians still play analog-based instruments and use analog mixers, effects components, amplifiers, *etc*.). The digital revolution has, and will continue, to effect modern society in profound ways.

### *2.2 The Interactive Artist*

The ubiquity of the computer and the proliferation of digital media has also ushered in a new type of artist – the interactive artist. The interactive artist uses the computer's capabilities for real-time autonomy and responsiveness to user inputs to explore new forms of expression, and uses the opportunities provided by digital technology to find deeper relationships with their audience. As Andrew Stern (2001) mentions: "By making the computer listen to the audience (the first half of reactivity), think about what it heard (autonomy), and then speak its thoughts back to the audience (the second half of reactivity), the artwork can have a dialog, a conversation, with the audience."

Who is the interactive artist? They are someone who knows how to think procedurally in order to express their artistic intentions. They are someone who has a strong foundation in artistic practice and information technology. They are an artist-programmer.

> "To create a computer-based artwork that captures the processes at work in a conversation requires programming. There is no escaping the fact that to make an artwork interactive is fundamentally to build a machine with processes; anything less would simply be a reactive work without autonomy – 'push button' art." (Stern 2001).

There is a danger, however, to the hype surrounding new media. The danger is that the interactive artist becomes so engrossed in what the new technology *can* do that they forget the underlying rationale behind the art itself. Children today are growing up with their appetites for agency fed by the explosion of the digital revolution. The world of the room-sized mainframe has been replaced by the omnipotent desktop computer connected to the information reservoir of the internet. The result is that media producers, academics, and artists alike must all now re-evaluate their role in a computationally generated landscape. For the interactive artist grappling with the problems inherent to digital technologies, such as continuity, granularity, and authorship (Strohecker 1997), it is important to remember the natural role of art "as a tool for exploring and critiquing relationship itself" (Rokeby, 2003b).

The purpose of this section is to emphasize my belief that to be successful at interactive cinema from the perspective of the researcher, producer, or creator, one must have a strong understanding of what it means to be an interactive artist. The interactive artist uses new media objects as the paint, and the computer screen as the canvas. The following section goes into a more detailed description of new media artifacts.

### *2.3 What is New Media?*

The concepts and ideas inherent to interactive cinema stem from the principles of new media, and so what follows is a brief discussion of what new media is and how it is represented. Coming up with a clear definition for new media is far from trivial. The most common understanding of new media, as implied by its digital nature, is that a computer is involved in some capacity, to either store information, produce it, represent it, or all of the above (Manovich 2001).

What **IS NOT** new media:

1. Print (books, magazines, newspapers)
2. Analog photography (cameras which use celluloid-based film)
3. Analog music (recording and playback)
4. Radio
5. Television
6. VCRs and VHS cassette tapes
7. Traditional cinema (using celluloid film stock as the recording medium)

Even the above list of what is not new media quickly runs into trouble as distinctions become blurred. Most TV shows, for example, are recorded on digital cameras, and yet the majority of subscribers still receive an analog television signal, so is television an old medium (as classified by the broadcast/cable signal) or new (as classified by the recording device)? Most movies are still shot using celluloid film stock and presented in theatres using analog movie projectors, and yet practically all modern movies are digitally edited and composited with computer-generated special effects. While the radio airwaves are dominated by analog broadcast signals, many radio stations are run digitally, with the functions of the mixing board, music playback, and advertisements all controlled by computer. Even books, magazines, and newspapers typically go through page layout programs on computers before being printed.

What **IS** new media:

1. Digital photography (digital still images)
2. Digital music (digital recording/editing)
3. Pay-On-Demand television
4. Digital cinema (digitally recorded and edited)
5. Interactive art (including interactive cinema)
6. Video games
7. Websites

8. CD-ROM multimedia titles

9. Virtual Reality (VR) environments

Factoring out the common features of the members of this (far from inclusive) list, certain principles begin to emerge. The more obvious principle being that any media form that is recorded or created digitally, and yet presented through an analog device, is still, in fact, new media. The next section discusses the principles of new media in further detail.

## *2.4 The Principles of New Media*

> *"A new media object may be a digital still, digitally composited film, virtual 3-D environment, computer game, self-contained hypermedia DVD, hypermedia Web site, or the Web as a whole"* (Manovich 2001, p.14).

In his book *The Language of New Media*, Lev Manovich (2001) outlines a number of unifying principles of new media. The following sections discuss these ideas.

### *2.4.1 Numerical Representation*

This first principle is another way of stating that new media objects are digital – they are described by numerical representations or computer code. At the lowest level, new media objects are essentially just information encoded into 1's and 0's (binary representation). New media is thus programmable and subject to mathematical manipulation. Digital objects can be created entirely within the computer, or created by transforming analog information into digital form. The analog-to-digital conversion process is called *digitization*, and involves two steps: (1) sampling at a particular resolution; and (2) quantification of the data into discrete units. Digital recording of a sound, for example, involves taking many discrete samples of the sound wave's instantaneous amplitude; the number of samples that are recorded per second refers to the sampling rate, and describes the resolution or detail at which the sound is recorded.

### *2.4.2 Reproducibility*

The binary representation of information allows for that information to be replicated without loss or degradation. This principle of new media, and the ability to distribute digital information over the World Wide Web, is what has sparked the current music copyright debate between record labels and file sharing protocols on the internet (e.g. Napster, Gnutella, Morpheus). The digitization of information, and the ability for perfect reproducibility, allows for new media objects to have theoretically infinite distribution potential.

### *2.4.3 Modularity*

*Modular* can be defined as "constructed with standardized units or dimensions for flexibility and variety in use" (www.m-w.com). For new media objects, the 'standardized units' are discrete packets of binary information[1]. With discrete representation comes the ability to break information into levels of granularity, or modules, with each level having the same structure as all the others. This is the fractal nature of new media, where a new media object can be composed of smaller new media objects, which are themselves composed of smaller new media objects, and so on. Later in this thesis I will show how this modular nature of new media can be exploited to create an object-oriented paradigm for interactive cinema.

### *2.4.4 Automation*

One thing that computers can do, and which they do exceedingly well, is perform simple repetitive tasks. The binary representation of data, and the modular structure of new media objects, enable "the automation of many operations involved in media

1)

[1] Analog is often referred to as something continuous, with digital referred to as something discrete.

creation, manipulation, and access" (Manovich 2001). In creating high-resolution computer-generated 3-D graphics, for example, the computer spends a great deal of time, independent of the user, executing complex mathematical calculations to render simple geometric forms into detailed visual images.

### *2.4.5 Dynamically Generated Content*

New media allows for content to be generated dynamically, specific for each user or group of users. A new media object can exist in any number of versions, and is not fixed or unchangeable. A website object, for example, such as a banner ad, could look different every time a user visits it through the dynamic creation of page content, or a particular version of a website could be seen based on the hardware (e.g. internet connection speed) and/or software (e.g. operating system, browser type) of the user. Internet browser 'cookies' can even be used to collect information about the user to create an individually tailored media experience.

## ***2.5 The New Media Interface***

Marshall McLuhan is famous for his dictum that "the medium is the message." This is especially true with new media objects, for they all need to have an associated interface in order to be manipulated, customized, and programmed, and yet any interface brings with it strong messages of its own.

Recognizing that all new media objects must be perceived through an interface, Lev Manovich argues that the old dichotomy, popular among traditional communication theorists, of *form vs. content* should be rewritten as *interface vs. content* (Manovich 2001). In reality, of course, it is difficult to truly separate form from content, or interface from content, but it is still useful to mention the dichotomy as a means of recognizing that the *form* of digital objects is not 'physical' – they are simply computer code accessed through an interface.

Why is the interface of new media objects relevant to the discussion of interactive cinema? It is relevant because many new media objects use cinematics to present themselves to the user. For a case in point, look at any website that relies heavily on Macromedia's Flash program to present its content[2]. Flash was designed with cinematics in mind, providing the author a frame-based timeline and excelling at creating animation for the web.

> "... the visual culture of a computer age is cinematographic in its appearance, digital on the level of its material, and computational (i.e. software driven) in its logic" (Manovich 2001, p.180).

One would be hard pressed to find a current video game that does not use cinematics as an introduction to the game, as cut-scenes, or as integrated components of the actual gameplay. When games first began employing cinematics, some of them used digital movies of live actors superimposed over a rendered graphical background; games are now moving towards using computer-generated characters that can be rendered and manipulated in real-time. This allows game designers to move from branching-type narratives (where every possible choice needs to be filmed – a very expensive and time consuming process), to dynamically generated content. Synthetic characters still look synthetic, however, so you do lose the realism generated from using live actors, but the distinction is narrowing rapidly[3].

### *2.6 A Definition for Interactivity*

The word *interactivity* has become a problematic term. It is used extensively to describe works of new media and interactive art, encompassing everything from simple

1)————————————

[2] See www.brokensaints.com for an example.

[3] See the movie *Final Fantasy: The Spirits Within* (2001) for a good example of highly realistic computer-generated characters.

hyperlinks to immersive virtual environments where the user navigates a 3-dimensional space with an infrared wand. *Interactivity* is used to describe the reading of a novel, the menu presented to a user when playing a DVD movie, and the experience players of a Massive Multiplayer Online Game (MMOG) have when they decide to go against the game's narrative spine and design their own quests. For this reason, interactivity needs to be understood as a term that encompasses many different levels of interaction.

In discussing the Japanese journalist and curator Itsuo Sakane, David Rokeby discusses how Itsuo considered any art to be interactive if we accept that viewing and interpreting a work of art is a form of participation (Rokeby 2002). Marcel Duchamp reiterates this as "the spectator makes the picture" (Duchamp in Rokeby 2002). The views of these authors illustrate an important point: even though interactivity is commonly used to refer to participatory digital technologies, its implicit meaning has broader applications.

Rokeby defines interactive art as a dialogue between the interactor and the digital artwork: "The interactive system responds to the interactor, who in return responds to that response" (Rokeby 2002). This refers to the concept of a 'feedback loop' that many people associate with interactive systems, and which I believe is a very effective way of creating agency. The extreme view is that a media object cannot be said to be interactive unless both the spectator/interactor and the media object are in some way changed permanently by each other or affected by the exchange (Rokeby 2002). In order for a media object to meet this requirement it would need to have enough intelligence to be able to adapt and grow according to its surroundings and user interactions. Limiting the definition to intelligent media or to where there is mutual transformation is disputable, however, as one can be said to interact with a book or magazine when they pick it up and turn the pages to read it. In a *Choose-Your-Own-Adventure* book (also called a G*amebook*), for example, the user navigates a multi-linear story by choosing to follow particular forking paths – the reader is periodically confronted with a story decision, and is directed to a specific page number based on their choice. A *Choose-Your-Own-Adventure* book is an analog form and yet is definitely interactive on some level.

Eric Zimmerman (2001) in his article "Against Hypertext" recognizes the broad usage of the term; he proposes four 'modes of interactivity' to help bring some clarity to the discourse:

### *2.6.1 Interpretative Interactivity*

Interpretative interactivity is cognitive participation with a media object. The interaction is with the 'content' of the media, such as the interpreted story from reading a book or watching a movie. This is the level which Itsuo Sakane was referring to, where the interaction is on the level of the "psychological, emotional, hermeneutic, semiotic, reader-response" (Zimmerman 2001). In this domain almost anything can be considered to be interactive as long as it involves some form of viewing and interpreting.

### *2.6.2 Utilitarian Interactivity*

Utilitarian interactivity is functional participation with a media object. This includes interaction with the physical, such as the weight and texture of a book, as well as the functional textual apparatuses, such as a table of contents or index. Although the manipulation of a mouse pointer on a computer screen is on a different level of interactivity, the physical movements of the mouse and button depression by the user fall under this category.

### *2.6.3 Designed Choice Interactivity*

Designed choice interactivity is explicit participation with a media object. This is interactivity in the sense that it is typically used when referring to new media. Included here are participatory actions such as clicking a hyperlink, working your way through an action-adventure game, experiencing a flight simulator, *etc*. This is a programmed interactive experience. Designed choice interactivity involves a two-way communication system between the media object and the user, and just as the user has the power to exert authority on the media, the media also has the ability to exert its influence in the exchange. In designed choice interactivity, the user receives feedback from the media object based on their input – but the media is not just *reacting*, it is also manipulating the user on a functional level (through programmed limitations, constraints, guidance, *etc.*) and on an emotional level (impacting a user's psychology). The user is thus transformed (psychologically speaking) by the media object, just as they themselves transform it (via the responsiveness and flexibility of the program).

When referring to interactivity throughout the rest of this thesis, I am referring to designed choice interactivity. The *interactive* in Object-Oriented Interactive Cinema (OOIC) refers to a relationship between a user and a cinematic media object where the user has agency over the object's form and/or content, and thus actively contributes to the construction of meaning and/or influences the media object's response(s). A standard novel would therefore not be interactive under this definition because the reader has no control over the book's form or content, whereas a *Choose-Your-Own-Adventure* book, while not digital, is still interactive because it allows for the user to co-construct meaning by deciding on their own story path through the branching narrative structure.

An interesting aside is that, using this definition, it could be said that one of the earliest examples of interactive art is a novel by Laurence Sterne called "The Life and Opinions of Tristram Shandy," finished in 1766 (Rokeby 2002). There are blank pages in Sterne's novel where the reader is intended to draw and write according to the narrative's flow, bringing the reader's subjective experiences into the weave of the story.

### *2.6.4 Macro-interactivity*

Macro-interactivity is cultural participation with a media object. Examples of this type of interactivity are fan culture; *The Sims* Exchange, a website where players of *The Sims* computer game can download playable game items (such as families, houses, skins, etc.) uploaded to the site by other players; the rise of popularity of the Laura Croft character from the video game *Tomb Raider* to become a sex symbol; and the in-game communities built up amongst players of Massive Multiplayer Online Games (MMOGs). Macro-interactivity is where "readers appropriate, deconstruct, and reconstruct... media, participating in and propagating massive narrative worlds" (Zimmerman, 2001).

### *2.7 Interactive Digital Media*

The list in Section 2.3 of what is new media encompasses a broad range of media forms. According to the principles described in Section 2.4, new media can be anything from a digital still image to a highly interactive and immersive 3-D virtual reality experience. As a means of subdividing the category of new media into more relevant domains, I propose the term *interactive digital media*[4] to refer to those new media objects which demonstrate traits centered around interactivity.

In her book *Hamlet on the Holodeck*, Murray (2000) outlines what she believes are the properties (procedural, participatory, spatial, and encyclopedic) of digital environments. I would argue that these are, in fact, properties of interactive digital media in particular, and pertain to a narrower domain than the encompassing sphere of 'digital environments.' I have chosen not to use the term *digital environments* in my discussion to emphasize the point that *interactive digital media* is a subcategory of new media, and that it describes properties that refer to particular new media *objects*. Interactive digital media characterizes artifacts which relate to a finer level of granularity (within the sphere of new media) than is implied by the word 'environment.' I also feel that Murray's term privileges *spatiality*; in my research I prefer to privilege the *procedural* and *participatory* aspects of new media. Nonetheless, Murray's text presents the exact four properties that I

1)————————————

[4]The *interactive* of interactive digital media refers to designed choice interactivity (see Section 2.6.3).

associate with interactive digital media, and thus the following discourse on the principles of interactive digital media is based on her framework.

### *2.7.1 Interactive Digital Media is Procedural*

In order to have coherency, interactive digital media objects have sets of rules that govern a user's experience. These rules are not meant to impose limitations, but rather to provide structure to what could otherwise be a chaotic digital landscape[5]. The procedural nature of interactive digital media implies that there are a series of steps that are followed according to a given set of instructions. The nature of interactive digital media is to allow for exploration of new possibilities and creativity through a procedural structure.

> "The most important element the new medium adds to our repertoire of representational powers is its procedural nature, its ability to capture experience as systems of interrelated actions. We are now engaged in establishing the building blocks of a procedural medium..." (Murray 2000).

### *2.7.2 Interactive Digital Media is Participatory*

To say that an interactive digital media object is participatory implies that the user is able to interact with the object on a functional level to induce a sense of agency[6] on the emotional level. One of the most compelling features of interactive digital media objects, enabled by the computational power of the computer, is the provision to the user to exert influence, to be an instrument in the design and/or manifestation of content. Having the ability to elicit a particular effect or change on an interactive digital media object turns the user from a passive observer into an active participant. Even though we are forced to

1)———————————

[5] Choas could, of course, be something desired by the interactive artist, but the procedural nature of interactive digital media would still necessitate a 'programmed' chaos.

[6] Agency, along with immersion and transformation, are referred to by Murray as the three pleasures (or aesthetics) of digital environments (Murray 2000).

work within a specific set of rules due to the procedural nature of interactive digital media objects, they are still quite appealing to us because of the agency they give us to create new experiences. "Agency is the satisfying power to take meaningful action and see the results of our decisions and choices" (Murray 2000, p.126).

Participation with an interactive digital media object means that the computer is in some way responding to the user; the program is not just displaying what the user inputs, but actually interacting in a form of dialogue. The computer game *Zork* is an early example of an interactive digital media story object. *Zork* enabled users to enter a fantasy world presented through words on the screen, where they could navigate the textual landscape by typing in commands which the program would respond to. In this way the user could manipulate objects, search dungeons for treasure, and fight off evil monsters in their quest to complete the adventure. What was so appealing about *Zork*, in spite of the fact that it provided no visual stimulation, is that its interactivity enabled users to feel a sense of agency over story.

The participatory nature of interactive digital media is a crucial feature, for one of its goals is to enable the active creation of belief, rather than just the passive suspension of disbelief.

### *2.7.3 Interactive Digital Media is Spatial*

Unlike books or print, interactive digital media has the ability to represent a space that can be traveled by the 'reader' beyond simple indexical navigation. "The computer's spatial quality is created by the interactive process of navigation" (Murray 2000, p.80). This ability to navigate a digital space helps instill within the user a sense of immersion:

> "The experience of being transported to an elaborately simulated place is pleasurable in itself, regardless of the fantasy content. We refer to this experience as immersion." (Murray 2000, p.98).

An interactive digital media object can have either open or closed navigation architecture. Open architecture means that the virtual space created by the interactive digital media object is free to be explored by the end-user – there are no specific boundaries guiding the user in a particular direction. Closed architecture is the opposite of open architecture, in that the end-user is forced to navigate a specific path. The web as a whole can be considered an interactive digital media object with open navigation architecture, whereas many computer adventure games employ a closed architecture, where if the user deviates, obstacles and *cul-de-sacs* are used to bring the user back to the main path.

#### *2.7.4 Interactive Digital Media is Encyclopedic*

Another way of stating this category is that interactive digital media is *database-oriented*, owing to the fact that digital media is able to store vast amounts of information. In computer science, a database is defined as a structured collection of data, or an organized body of related information, and in digital media, the database has emerged as an effective medium to provide a user efficient access to information.

There are many different types of databases – hierarchical, network, relational, object-oriented, *etc.* – and they each use different models and algorithms for organizing the data. The Object-Oriented Interactive Cinema (OOIC) model uses an object-oriented database, which stores complex data structures (objects) in hierarchical classes.

### ***2.8 What is Interactive Digital Media?***[7]

The following is a list of some examples of interactive digital media. All of these also appear on the 'What is New Media' list since interactive digital media is a subset of new media.

1. Interactive art (including interactive cinema)

1)

[7] See Figure 41 in the Appendix for a schema on how the domains of new media and interactive digital media, as well as the OOIC model, fit in with their related media spheres.

2. Video games
3. Websites
4. Interactive CD-ROM multimedia titles
5. Virtual Reality (VR) environments

As can be seen by this list, *interactive cinema* is an example of what I am calling interactive digital media, and thus it inherits all of Murray's principles described in Section 2.5 (i.e. interactive cinema is procedural, participatory, spatial, and encyclopedic).

### *2.9 The Degree of Interactivity*

A major problem with interactive media based on designed choice interactivity is that the more interactive the content, the more labor required to create it. But the greater the interactivity, the greater the user's sense of agency.

One argument is that a media object with very limited interactivity can actually be less satisfying than had it had no interactivity at all, for simple hyperlinking is essentially asking the user to follow a set of pre-programmed subjectively existing associations. With only a minimal amount of interactivity, the programmer's mind is exposed, thus making it harder for the user to mistake the programmer's cognitive processes for their own. This is thus an argument that interactive media should be extensively interactive or not interactive at all (in which case it is up to the user's imagination to go beyond the embedded content).

The other argument, however, is that users do not need a high level of interactivity to be satisfied, and that too many choices can actually be detrimental to the user's interactive experience, making the increased labor and production costs to the media creator hardly seem worth it. An overly interactive work, for example, can generate in the user feelings of confusion, lack of focus, lack of purpose, and/or hinder suspension of disbelief. Wimberley and Samsel argue that a limited branching structure, if done well, can be all that is needed to satisfy a user's pursuit for interactivity:

> "Hyper-interactivity will not lead to audience satisfaction; it will lead to betrayal, a scenario made inevitable by the tyranny of choice... As a structure, branching doesn't allow for a high degree of end-user interactivity. It does, however, fulfill a key element of interactive design – user choice." (Wimberley and Samsel 1996).

The choice by the interactive artist for the degree of interactivity is thus a balancing act between the desire to provide their audience with agency, the desire to have artistic control over how their content is presented, and the amount of time and expertise available for programming interactivity into the system.

### *2.10 Why Be Interactive?*

For what purpose am I attempting to bring interactivity into the cinematic experience?

> "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." (Galyean 1995).

Interactive media is often perceived as an externalization of the mind, of making the author's cognitive process a part of the public knowledge domain. As users we enjoy interactive experiences because, if done well, they give us a heightened sense of agency by allowing us to customize our experience. We enjoy agency because we like to make a difference, exert control, and see the consequences of our actions.

Agency, for example, is an important feature of hyperlinking on the internet:

> "The very principle of hyperlinking, which forms the basis of interactive media, objectifies the process of association, often taken to be central to human thinking" (Manovich 2001, p.61).

Interactivity also allows for immersion, which is to say that it leads the interactor into a more engrossing and absorbing experience. Through immersion, the interactive artist hopes to reach their audience on a deeper, incisive, and more penetrating level. The designer's world becomes that of the user's.

Through effective elicitation of agency and immersion, interactive cinema is "capable of expanding the social engagement of audiences while offering intensive narrative immersion in a story experience..." (Davenport *et al*. 2000, p.456). Interactivity need not be limited to hyperlinking or button clicking, of course. Interactive media can involve the capturing and interpretation of body movements, gesture, light sources, proximity sensors, *etc*. The key obstacle with this kind of 'physical' interactivity is how to effectively capture the user's input and then appropriately and intelligently interpret the signals to produce a meaningful experience. This is one of the goals of the Object-Oriented Interactive Cinema model, as the framework advocates increased immersion through non-standard user interfaces (i.e. something other than a mouse and keyboard). This aspect of the OOIC model is discussed further in Chapter 9.

# Chapter 3:
# Interactive Narrative

> *"[T]he computer can be a compelling medium for storytelling if we can write rules for it that are recognizable as an interpretation of the world"* (Murray 2000, p.73).

Interactive narrative is a new kind of storytelling. It combines the technical with the artistic into a new sphere in interactivity design, and requires that the writer take a far more abstract approach than they would with traditional storytelling (Crawford 2000). To experience a story at a new level, "we need to understand how a viewer can participate in a drama in interesting and engaging ways without disrupting the plot" (Galyean 1995). It is often considered that one must sacrifice narrative for increased interactivity, or that one must give up some interactivity to preserve effective narrative. But does a lack of linearity diminish the effectiveness of a narrative? Proponents of hypertext do not believe so:

> "In a hypertext environment a lack of linearity does not destroy narrative. In fact, since readers always, but particularly in this environment, fabricate their own structures, sequences or meanings, they have surprisingly little trouble reading a story or reading for a story." (Landow 1997).

This chapter discusses the interactive narrative, which is an important component of Object-Oriented Interactive Cinema.

### *3.1 What is Narrative?*

To go beyond the traditional boundaries of narrative, we must first have a clear understanding of the terms *narrative*, *story*, and *plot*. The task of defining these terms, however, is difficult and extensive. Indeed, in my research into narratology I have come across numerous, and sometimes contradictory, definitions for these terms as proposed by various narrative theorists (Chatman 1993; Gander 1997; Forster 1927; Ryan 2001), but was unable to obtain a clear consensus. Therefore, for the purpose of consistency I have decided to use the concepts and terminology from Bordwell and Thompson's (1997) *Film Art: an introduction*. These definitions are the most useful for my research because of their clarity and unambiguity, and also because of the authors' cinematic backgrounds.

### *3.1.1 Narrative*

A narrative tells a story. It is a series of events occurring in time and space, with the events typically linked through cause and effect relationships. In fact, many narratologists believe that in order for a narrative to be meaningful, it *must* consist of logical cause and effect relationships organized in time:

> "Narrative representation must be thematically unified and logically coherent. Their elements cannot be freely permuted, because they are held together in a sequence by relations of cause and effect, and because temporal order is meaningful." (Ryan 2001).

I believe that this definition holds true for most cases, but not for experimental filmmaking (or experimental art in general), where the artist may, in fact, desire a lack of logic or cause/effect relationships. Here the artist generates meaning through the purposeful disregard for causality (i.e. an abstract narrative). A narrative is still created, although one outside of conventional definitions. What this implies is that on some level, almost anything can be considered a narrative, given a plot and story, whether it is logically coherent or not – it depends on the reader's interpretation and experience of the work. "It is often argued that narrative... [is] something that is only discovered or reconstructed by the reader after the end is reached..." (Aarseth 1997). In other words, for a narrative to be meaningful, it needs to make cognitive sense to the reader.

Within the narrative a *pattern* may be found. A narrative pattern typically links the main story elements together in a holistic theme. Many writers and narratologists believe that there are a discrete number of possible narrative patterns, "corresponding to the basic patterns of desire, fulfillment, and loss in human life" (Murray 2000). The belief is that any story created can be placed into one of these categories. For example, Ronald

B. Tobias suggests that there are twenty 'master' patterns in all of literature (Tobias 1993):

- Quest
- Adventure
- Pursuit
- Rescue
- Escape
- Revenge
- The Riddle
- Rivalry
- Underdog
- Temptation
- Metamorphosis
- Transformation
- Maturation
- Love
- Forbidden Love
- Sacrifice
- Discovery
- Wretched Excess
- Ascension
- Descension

This qualification lends itself to computationally generated narratives, provided that each category can adequately be described and sets of rules or qualifiers assigned. The premise is that since a narrative is a patterned set of associations, it can be described by a program. "The formulaic nature of storytelling makes it particularly appropriate for the computer, which is made for modeling and reproducing patterns of all sorts" (Murray 2000). No one would want to hear a story that was a random generation of narrative patterns, however, and so the challenge is to tell the computer what patterns to use and how to use them. The first step is to see a narrative as a collection of smaller narrative elements. As Brooks states when referring to story-generation programs, "a storytelling system is not a magic box which creatively makes up a story when asked... but a system of specially stored and organized narrative elements which the computer retrieves and assembles according to some expressed form of narration" (Brooks, 1996, p.318).

Narrative can thus be divided into *plot* and *story* to distinguish those elements which are explicitly presented to us (*plot*), and the set of all events that the reader creates in their own mind from the presented plot elements (*story*) (Bordwell and Thompson 1997). In the OOIC model, the movie objects that are presented to the user (based on the user's interaction with the system) would make up the plot, and the resultant chronicle and experience that developed out of the user's interpretation of these objects, the story[8].

The process can be summarized as follows:

1. The author, who has a particular story that they wish to present, builds an interactive system based on the OOIC model.
2. The user interacts with the system, and based on their social and cultural influences, the user makes choices. The system in turn, records and acts upon these choices.
3. Based on the user's interactions with the system, a plot is created (consisting of that which is presented to the user).
4. The user witnesses the plot and constructs a story in their minds based on the experience.

The following two sections discuss *story* and *plot* in more detail.

1)

[8] See Figure 42 in the Appendix for a diagrammatic flowchart of the sequence of events leading from author to story under the OOIC model.

### *3.1.2 Story*

> "Story is... frequently used as a reference to some general or abstract description of a meaningful collection of events, people and/or things... Story, then, can be thought of as a system of associations between elements, composed of events, people, and things." (Brooks, 1996, p.317)

A story is a collection of events in time – a chronology of events. The story world, sometimes called the *diegesis* (the Greek word for "recounted story"), is what the reader creates in their mind from the presented story events. The story is comprised of both those events that are explicitly presented, and those events that the reader infers but which are not actually found within the presentation itself (Bordwell and Thompson 1997).

Any piece of art can be said to have a story, for just as it takes time to read the lines of a book or poem, it also takes time to view a painting or sculpture, listen to a piece of music, or appreciate a modern dance performance. These can all be seen as the experiencing of a chronology of events, and the story is what the audience creates in their mind from the experience.

### *3.1.3 Plot*

The plot describes "everything visibly and audibly present" (Bordwell and Thompson 1997, p.92). The plot is made up of a narrative's explicitly presented events as well as the *nondiegetic* material (i.e. content that is external to the story world).

Plot and story overlap in that the plot shares the explicitly presented events with the story, but the story expands on these by adding inferred events which are never actually presented to the reader. Likewise, the plot goes beyond the story by presenting *nondiegetic* events, such as film credits, a musical score, or a book's title and index, but which still affect the narrative experience. (Bordwell and Thompson 1997).

Within the plot can be found techniques through which the author manipulates the narrative elements that are presented to the reader. Plot manipulations refer such narrative devices as similes, metaphors, verse, prose, *etc*. In cinema, the plot manipulations also include framing, cutting, dissolves, camera movement, camera angles, point of view, *etc*. (Felluga 2003).

### *3.1.4 The Aristotelian Arc*

For a story to be compelling it generally requires the generation of tension and release. The idea of building up a story bit by bit to a crescendo and then tapering off in a *denouement* dates back thousands of years. The Greek philosopher Aristotle wrote about it in his famous book *Poetics*, and as such the traditional narrative curve is often referred to as the Aristotelian Arc.

The Aristotelian concept of narrative is typically constructed from (1) Set-up, (2) Complication, (3) Development, and (4) Resolution. In the set-up phase, the stage is set and the initial state of affairs established. The complication is where the action changes direction, or when the principle obstacle is encountered that is preventing resolution. In the development stage the narrative builds on the established premises and goals, often by creating tension and suspense. Here is where the main incidents occur, eventually leading up to the climax. In the climax, or resolution, there is a turning point that intensifies the action and often brings into question whether the complication will be resolved or not.

Many writers have expanded and/or further subdivided these four categories, but the fundamental concept remains the same. Edward Branigan (1992), for example, proposes the following narrative scheme:

1. Introduction of setting and characters.
2. Explanation of a state of affairs.
3. Initiating event.
4. Emotional response or statement of a goal by the protagonist.
5. Complicating emotions.
6. Outcome.
7. Reactions to outcome.

These seven categories, the categories defined by the Aristotelian Arc, or any other user-defined categories, can be used in the OOIC model as part of a meta-tagging system. The meta-tagging system functions by attaching particular narrative information to the film object, which can then be used by the program for sorting, ordering, searching, *etc*.

The concept of the narrative arc emphasizes a narrative built around causality. But what exactly is meant by *causality*?

> "Focused causal chains are not just sequences of paired story events in time and space, but embody a desire for pairing events and the power to make pairs. Narrative causes are thus principles of explanation, or criteria for grouping elements, which are derived from cultural knowledge as well as from physical laws…" (Branigan 1992).

Although a narrative based on causality is common to most stories, causally linked events are not a narrative requirement. In other words, a narrative is still possible without there being specific cause-effect relationships, but in this case the story is considered to be more of an abstract creation.

## *3.2 The Interactive Narrative*

### *3.2.1 Expanding the Definition of Narrative*

Perhaps the first step in determining what an *interactive* narrative is, and how one can be effectively created, is to expand our idea of narrative, or at least to not feel confined to standard preconceptions. As Grahame Weinbren mentions in "In the Ocean of Streams of Story," by introducing a user who has impact over the story, it forces us to "question the concepts of end and beginning, of crisis and conflict, of development itself" (Weinbren 1995). The traditional Aristotelian narrative arc of set-up, complication, development, and resolution, need not be completely disregarded, but perhaps we need to start looking at the arc in a different way – an oblique strategy. Instead of an overarching narrative arc that encompasses the entire story and experience, perhaps a multi-linear narrative could employ micro-narrative arcs contained within story fragments. Writers of interactive narrative could search for ways "to structure a coherent story not as a single sequence of events but as a multiform plot open to the collaborative participation of the interactor" (Murray 2000).

### *3.2.2 The Multiform Narrative*

A multiform narrative is one which has many different forms, shapes, or appearances, taking full advantage of the principles of interactive digital media discussed in Section 2.7, and combining a database of content with procedures for interaction and navigation. Producing a multiform narrative requires a rethinking of authorship.

> "... [W]e must reconceptualize authorship... and think of it not as the inscribing of a fixed written text but as the invention and arrangement of the expressive patterns that constitute a multiform story" (Murray 2000).

## ***3.3 Types of Interactivity***

This section outlines various approaches to qualifying interactivity, culminating in a table in Section 3.3.5 which offers an interactivity framework for Object-Oriented Interactive Cinema.

### *3.3.1 Some Common Approaches to Interactive Narrative*

- **Branching Narrative:** In a branching narrative, the story follows forking paths based on user interaction. Also called *branching-type* interactivity, or *menu-based* interactivity, information is used by the computer program to create interactivity based on the user's cognitive processes rather than body position, gesture, *etc*. Because of the potential for an exponential explosion of narrative paths as more and more branch-points are added, projects that employ this type of design tend to limit the number of possible paths available to the end-user. While this may work decently with hypertext or *Choose-Your-Own-Adventure* novels, it is not generally considered a good approach for multi-linear visual media (Davenport 1997; Strohecker 1997).

> "Both audiences and producers have found [plot-branching] dissatisfying. Directors find costs associated with the complex production to be prohibitive, and CD-ROM users and theatre-goers often feel they are missing something if they can't see all the possible endings." (Strohecker 1997, p.377).

- **Exploratory Narrative:** The user explores a virtual world, discovering narrative fragments as they progress, to gradually re-construct the whole story in their mind.
- **Generative Narrative:** The premise behind generative or automatist storytelling is that the structure of a story can be an emergent property of the interaction of the individual with a storyteller system.

### *3.3.2 Interactive Flow*

Chris Crawford discusses narrative and interactivity using the concept of 'flow' to refer to the number of choices that are available to the end-user. The number of possible avenues open to pursue determines if the flow will be highly branching or mostly linear. As Crawford asserts: "One way to judge the interactive quality of a design is to examine the ratio of accessible states to conceivable states" (Crawford 2000). In making a tree diagram of the possible paths available to the end-user, a more interactive work would be 'bushier' by having more branching paths, and thus more choices. Crawford is quick to point out the problems with a highly branching narrative, however. To illustrate this, Crawford asks us to consider an interactive story with one hundred layers, or action/event levels, where each node or branch-point has only two choices available to it (the minimum required). This would give us a total of 2 to the power of 100 nodes. How big is that?

> "If you employed every human being on this planet to create nodes, each person making one node every second, working 24 hours per day, 365 days per year, then it would take 5 trillion years to make the nodes necessary to build that single storytree. We seem to have a problem here." (Crawford 2000).

One work-around to explicit branching is to employ the 'foldback' technique, where the storyline folds back on itself through merging nodes. The problem, of course, is that the interactivity loses meaning, and the user loses some sense of agency, since no matter what the user does they will eventually come to the same place in the narrative. The foldback technique prevents true customization of the experience, as each user must follow the same story spine in order to proceed through the narrative. "If we wish to offer our user a truly satisfying interactive experience, it is imperative that we allow each user to express his/her individuality during the experience" (Crawford 2000).

Another approach is the 'kill them if they stray' technique, where "the designer allows the user many options, but almost all of the options lead to the death of the user or the termination of the story" (Crawford 2000). Many games, most notably adventure and puzzle games (e.g. *King's Quest*, *Tomb Raider*), use this technique, and although it may enhance the user's sense of agency, the story is still essentially presented through a linear discourse.

### *3.3.3 Open/Closed Interactivity*

Lev Manovich refers to the interactivity that uses fixed elements arranged in a branching structure as *closed* interactivity; interactivity where both the elements and the structure are generated dynamically in real-time based on user interaction he calls *open* interactivity (Manovich 2001).

Eric Zimmerman refers to closed interactivity as a content-based or embedded structure, such that the "content is already embedded in the system before any interaction begins" (Zimmerman 2001). An embedded narrative structure means that the content is pre-existing; no new content is created during the user's interaction that is outside of the non-linear narrative's program. Here the world is finite and there are a quantifiable number of possibilities available to the end-user.

Zimmerman refers to open interactivity as a system-based or emergent structure, such that the interaction develops from rule-sets and procedures to produce unexpected experiences and content (Zimmerman, 2001). In an emergent structure the world is not finite, and the end-user is able to, at least to some degree, co-construct the narrative. In this scenario the end-user can actually choose not to follow the narrative spine at all, but rather define a personal set of goals. There are a near-infinite number of possibilities available to the end-user. Under this division, the computer game *Myst* is considered more embedded and less emergent, whereas the simulation game *The Sims* is more emergent and less embedded. The open/closed or emergent/embedded dichotomies represent the ends of a continuum, however, and are not meant to be taken as discrete measurements for interactivity.

In *Hamlet on the Holodeck*, Janet Murray uses similar terms when discussing the spatial property of digital environments, referring to *open* and *closed architecture* (Murray 2000). Unlike books or print, digital realms have the ability to represent navigable space. A non-linear work with open architecture means that the virtual space is free to be explored by the end-user; there are no specific boundaries guiding the user in a particular direction. Closed architecture, on the other hand, is where the end-user is forced to navigate a specific path. If the user deviates, obstacles and *cul-de-sacs* are used to bring the user back to the main path.

Wimberley and Samsel define open and closed architecture as follows:

- Open Architecture:

  "An application design structure which permits the end-user to navigate freely from one path, event, scene or location to another, no matter where those choices may appear in the hierarchy of the program... Physical transportation through time or space is limitless." (Wimberley and Samsel 1996).

- Closed Architecture:

  "An application design structure which insists that the end-user choose a path, event, scene or location by retracing the current path until they reach a point in the program which offers the user a new event, scene or location to choose from... There are very specific boundaries limiting [the] end-user's world... [The] user must navigate within a branching or *Cul-De-Sac* path structure. [The] critical path must be achieved to move through the story's hierarchy." (Wimberley and Samsel 1996).

### *3.3.4 A Classification System for Interactivity*

Marie-Laure Ryan, in her essay "Beyond Myth and Metaphor," explores interactivity from a broader, completely user-centered perspective. Ryan sets up two pairs of opposing concepts: internal/external interactivity, and exploratory/ontological interactivity (Ryan 2001).

- In **internal** interactivity "the user projects himself as a member of the fictional world" (Ryan 2001). This can be done through an avatar or by experiencing the world from a first-person perspective (e.g. a first-person shooter video game). This is a form of *personal perspective* interactivity.
- In **external** interactivity the participant is situated outside of the virtual world, either by playing a god-like role, looking on and controlling from above, or by navigating a database story structure. This is a form of *impersonal perspective* interactivity that does not require a concrete persona in the virtual environment.
- In **exploratory** interactivity the user is free to move about the virtual world and explore all the details, but their activity does not affect the overall plot in any way. The user simply reveals the story as they travel around the virtual landscape.

- In **ontological** interactivity the decisions of the user can affect the plot, such that the story develops directly from the user's interactions.

Cross-classification of these two dichotomies leads to four combinations for categorizing interactive works. These are not listed in any particular order of effectiveness.

- **External/exploratory interactivity:** Certain 'classical' hypertexts fall into this category, such the 'novels' of Michael Joyce, Stuart Moulthrop, and Mark Amerika. Under this category the user can choose paths through the virtual space, but the space itself is unalterable. This type of interactivity is exploratory because "the reader's path of navigation affects not the narrative events themselves, but only the way in which the global narrative pattern... emerges in the mind" (Ryan 2001). The CD-ROM title *Ceremony of Innocence* is an example of external/exploratory interactivity, where the user is peripheral to the narrative, and reveals the story in steps by solving puzzles with the mouse pointer. Real-time strategy (RTS) games, such as *Command & Conquer*, *Age of Mythology*, and *Warcraft*, as well as some simulation games such as *Rollercoaster Tycoon*, fall under this category. In these games the user is never associated with an avatar in the narrative, and they control all of their 'troops' or 'creatures,' or manage the simulation they are building, from a completely external frame of reference. These games are exploratory for although the user has many freedoms and manners in which to solve the game's quests, the plot itself is linear, and is only revealed when the user completes a level and can move on.
- **Internal/exploratory interactivity:** In this category, "the user takes a virtual body with her into the fictional world, but her role in this world is limited to actions that have no bearing on the narrative events" (Ryan 2001). Even though the user cannot affect the story, they are still immersed in the narrative through a virtual body. They are present on the story's stage, if only as an observer.

Examples of this type of interactivity are the computer games *Riven, Tomb Raider, Diablo* and *Grand Theft Auto III*. In these games the user plays an internal role (e.g. Laura Croft in *Tomb Raider*), but the game is structured in such a way that the user must follow a fairly linear narrative path. Deviating from the 'correct' path simply stalls the narrative; the story does not progress any further until the user solves the puzzle/obstacle/mystery to open the narrative gateway.

- **External/ontological interactivity:** In this category the user is outside of the virtual world, but has some control over the plot and the fates of the characters, such that the story is generated directly from the user's decisions. Typically in this case, because the user is external to the narrative, "the individual forking paths in the plot are... less interesting than the global pattern of their interconnections" (Ryan 2001). Computer games that employ god-based interactivity, where the user plays an all-powerful being controlling the world's destiny, usually fall into this category (e.g. *Populous, SimCity, The Sims, Caesar*). Many of these games attempt to increase immersion by giving the user a character to play in the game. For example in *Caesar* you play a Roman Emperor governing your city, and in *SimCity* you are the mayor. Although these characters are not physically represented in the games themselves, it could be argued that this contextualization of the user's role places these games somewhere between external/ontological and internal/ontological on the interactivity continuum.
- **Internal/ontological interactivity:** In this category the user is internal to the narrative, and has the ability to control their own fate by making decisions according to the laws of time and space of the virtual world. If the *Holodeck* of the Star Trek universe was ever to be implemented, this is the category under which it would fall. This category is actually quite encompassing, however, as *Choose-Your-Own-Adventure* books are also an example of this type of interactivity. Reading these books, a user flips through the first-person text to

follow a branching-type narrative, ultimately governing the path that they will take through the different narrative possibilities. A good example of internal/ontological interactivity that is popular today is the Massive Multi-Player Online Game (MMOG), where thousands of players interact over the internet in a shared virtual world. Players are actual characters (represented by avatars) who travel around and explore the fictional realm, and who can either choose to follow the native storyline established by the game creators, or diverge from this and create their own sets of goals and activities, thereby creating a generative narrative. In one MMOG, a group of players got together and, instead of following the established game quests, decided to create their own club (called a guild) with no other purpose than to socialize with each other. For these people, the narrative experience emerged from their own interactions and relationships.

#### *3.3.5 An Interactivity Framework*

I believe that Ryan's four categories of interactivity are very useful, although I do have difficulties with her treatment of ontological interactivity. According to Ryan's framework, a branching-type narrative that has a fixed, and often quite limited, number of possible paths can be in the same category as the *Holodeck* model, where the experience is completely generated through user interaction and there are no fixed traversals through the narrative content. I also question the appropriateness of the term *ontological*, as ontology is defined as "the metaphysical study of the nature of being and existence" (www.dictionary.reference.com). A more appropriate term, I believe is *transformative*, which can be defined as "having power, or a tendency, to transform," and which speaks more to the mutability of the narrative structure of this category.

For these reasons I believe that *ontological interactivity* needs to be deconstructed and changed to **embedded transformative interactivity** and **emergent transformative interactivity**. As mentioned in Section 3.3.3, *embedded* and *emergent* are Zimmerman's terms for the *open* and *closed* terminology used by Manovich, Murray, and Wimberley and Samsel. Under this framework a *Choose-Your-Own-Adventure* book, as well as games like *Sacrifice* which have some plot branching, are examples of **internal/embedded transformative interactivity**, and a Massive Multi-Player Online Game is an example of **internal/emergent transformative interactivity**. The DVD movie *Point-Of-View* and many branching-based hypertext narratives are examples of **external/embedded transformative interactivity**, whereas examples of **external/emergent transformative interactivity are** games like *The Sims*, *SimCity 4*, and any other god-game that has a large number of possible narrative paths.

In summary, the framework for interactivity which I use for the Object-Oriented Interactive Cinema model can be represented by the following table[9]:

1)

[9] See also Figures 43 and 44 in the Appendix for two representational models of the OOIC interactivity framework.

| <u>**Interactivity Type**</u> | <u>**Description**</u> | <u>**Examples**</u> |
|---|---|---|
| External/exploratory | The user, who is external to the narrative, can choose paths through the virtual space, but the space and narrative are unalterable. | Classical hypertexts, *Ceremony of Innocence*. |
| Internal/exploratory | The user cannot affect the plot, but they are still immersed in the narrative through a virtual body. | *Riven, Tomb Raider, Diablo Grand Theft Auto III*. |
| External/embedded transformative | The user is outside of the virtual world, but has some control over the plot and/or the fates of the characters, such that the story is generated directly from the user's decisions. | *Point-Of-View* DVD, branching-based hypertext narratives. |
| External/emergent transformative | The user is outside of the virtual world, but has a high degree of control over the plot and/or the fates of the characters. | *The Sims*, *SimCity 4*. |
| Internal/embedded transformative | The user is internal to the narrative, and has the ability to control their own fate to a limited extent by making decisions according to the rules of the virtual world. | *Choose-Your-Own-Adventure* books, *Sacrifice* |
| Internal/emergent transformative | The user is internal to the narrative, and has the ability to control their own fate extensively by making decisions according to the laws of time and space of the virtual world. | Massive Multi-Player Online Games (MMOGs) such as *EverQuest* and *Asheron's Call*. |

**Table 1:** Interactivity framework for the Object-Oriented Interactive Cinema model

## *3.4 Interactive Narrative Examples*

### *3.4.1 Video Games*

> "... once computer-based story and art become as deeply interactive as computer games are now, perhaps they will eclipse games as the dominant interactive format" (Stern 2001).

Video games present one of the most prominent examples of interactive storytelling in today's new media environment. Game consoles, such as the Sony *Playstation 2* and Microsoft's *XBox*, have become immensely popular with young to middle-aged males. Computer games are equally as popular as more and more people purchase personal computers. The ability to compete against other players over the internet has become practically standard with all new computer game releases. Most games do not, however, take advantage of the new possibilities for storytelling offered by digital environments. “So far, video games are the only form of electronic entertainment that have successfully shaken the general public’s notion of television viewing; however, video games rarely make us gasp with fear or tug at our heartstrings” (Davenport and Murtaugh 1997).

Games are usually categorized according to the nature of their gameplay: action/adventure, first-person shooter, real-time strategy, simulation, god-game. The narrative can likewise usually be described using simple phrases. “The story line in most gaming software can be described in terms of two or three morphemes (fight bad guy, solve puzzle, die)” (Murray 2000). Adventure or quest games, such as MUDs (Multi-User Domains, or Multi-User Dungeons), often rely on the repetition of plot modules focused around action sequences such as fight, explore, negotiate, socialize. The limited number of plot segments makes it easier to construct a compelling narrative, even if the narrative is fairly stereotypical. “The MUDs offer extensive opportunities for participation in formulaic narrative environments, but the collectively generated stories are diffuse and repetitive” (Murray 2000).

As many games today have a distinct focus on graphics quality, it is simply not possible for them to deviate too far from the main story spine due to the labor, production costs, and memory demands associated with a high degree of story branching. “The technological resources of the game-makers are directed toward rapidly transforming visuals rather than expressive storytelling” (Murray 2000). Perhaps this focus will shift if audiences eventually become less obsessed with stimulating visuals and more concerned with emotional resonance, but perhaps the focus relates more to how we treat the representation of story in a digital environment. “Games are limited to very rigid plotlines because they do not have an abstract representation of the story structure that would allow them to distinguish between a particular instantiation and a generic morpheme" (Murray 2000). Interestingly enough, Murray's use of *abstract representation* and *instantiation* overlaps with words or phrases found in an object-oriented vocabulary.

### *3.4.2 The Erasmatron*

The *Erasmatron* is a commercially available multi-linear storytelling tool developed by Chris Crawford. The *Erasmatron* uses a storytelling engine consisting of a cast of actors and a large database of verbs, which the actors can use in response to given dramatic situations (Crawford 2003). "The developing story is presented to the user through an interface, which uses a combination of text and a facial display capable of showing more than a hundred different facial expressions on any actor's face" (Crawford 2003). While the *Erasmatron* is an interesting tool, and may help a less programmatically inclined writer to compose interactive narratives, it only helps to build branching-type narratives within the external/embedded transformative interactivity category. The author is limited by the program, for example, to up to 30 actors, a limited number of personality traits, 62 settings, and 126 items (Crawford 2003).

Similar to OOIC, which manipulates and assembles a database of new media objects, the *Erasmatron* functions by assembling fundamental dramatic components (verb, subject, object, *etc*.) into various combinations. The OOIC model and the *Erasmatron* differ considerably, however, in that OOIC's focus is on the construction of a cinematic experience, and the *Erasmatron* is more of a literary tool, focusing on the sequencing of text with a supportive visual component. As discussed in Chapter 8, the OOIC model also suggests methods for adding stochastic processes to the interactivity design to add emergence to the otherwise strictly embedded narrative.

### *3.4.3 Other Examples*

Video games are certainly the most dominant and economically successful examples of interactive media today, but other formats are still finding niche audiences. Web soaps have been around practically since the dawn of the internet as writers quickly realized the potential of hyperlinking for multi-linear storytelling, but more often than not, the "link-happy exhibitionism of the Web soaps send us hopping from screen to screen in search of a coherent story" (Murray 2000).

CD-ROM narrative titles, such as *Ceremony of Innocence*, have gone the furthest in offering a filmic story experience, combining a compelling plot with audio and visuals. The typically limited navigability and essentially linear story unfolding, however, can often frustrate the audience's desire for agency.

Along with *Choose-Your-Own-Adventure* books, Julio Cortazar's *Hopscotch* is a good example of non-linear fiction. *Hopscotch* can be read from front to back like a conventional novel, or in an alternate order as proposed by Cortazar, offering the reader two different yet complementary approaches to the story (Costikyan 2003).

A number of researchers are looking into interactive storytelling systems, drawing on examples from theatre and narratology. Marie-Laure Ryan is exploring story-generation systems that divide a narrative into tree-based hierarchies, and supports her work using narratology theory, literary theory, discourse analysis, cognitive psychology and artificial intelligence (Ryan 1991). Brenda Laurel, who sees in the computer parallels to a theatrical environment, "has proposed an interactive fiction system presided over by a playwright who would shape the experience into the rising and falling arc of classical drama" (Murray 2000). Michael Lebowitz has created a storytelling system that assembles together morphemic segments derived from daytime soap opera plots: amnesia, murder, threats, forced marriage, and adultery (Lebowitz 1984). The principle obstacle in all of these ambitious systems is to provide the computer program with enough intelligence to be able to construct convincing tension/release narrative arcs.

# Chapter 4:
# Digital Cinema

Cinema has been revolutionized by the advent of digital technology. It is rare to find a Hollywood movie released today that has not been digitally manipulated in some capacity, whether through digital compositing, the addition of 3-D graphics, or by having been digitally edited using software such as Final Cut Pro, AVID, Adobe Premiere, Adobe AfterEffects, and/or ProTools. The digital revolution has also put the power to make movies into the hands of anyone with a video camera and a computer.

Object-Oriented Interactive Cinema is a form of digital cinema, and so this chapter serves to define digital cinema and discuss its properties.

## *4.1 What is Digital Cinema?*

Lev Manovich (2001, p.301) defines digital cinema using the following equation:

*Digital film = live action material + painting + image processing + compositing + 2D computer animation + 3D computer animation*

This definition implies that digital cinema is a "particular case of animation that uses live-action footage as one of its many elements" (Manovich 2001, p.302), but I cannot fully agree with this statement. In my opinion, *Toy Story* (1995), the first feature-length animated movie created completely using computer graphics, is very much under the category of cinema, and yet does not contain any live-action footage. I consider *Toy Story* to be the first example of digital cinema. Manovich, in contrast, refers to *Star Wars: Episode I – The Phantom Menace* (1999) as the first all-digital film because it uses real actors and sets in combination with computer graphics (Manovich 2001).

I do not fully support the distinction between a movie that references traditional filmmaking while using computer technology (e.g. *Star Wars: Episode I*) from a movie that is solely a product of computer technology (e.g. *Toy Story*). *Toy Story* uses camera angles, pans, tilts, zooms, and all the other techniques familiar to traditional filmmaking.

Most importantly, its discourse is cinematic. Perhaps Manovich's equation could be modified such that live-action material is an optional requirement.

For the purpose of this thesis, I am defining digital cinema as a *modus operandi* that uses digital technology for every step of the filmmaking process. This implies recording the movie on a digital camera, downloading the footage into a computer, and editing the movie using a digital non-linear editor. Even the audio is edited digitally. In this process all traditionally analog steps are bypassed or replaced with the digital equivalent; the only analog-to-digital conversions are those done by the digital camera when it captures a scene (light and color values are converted to code) through its one or more CCDs (Charge-Coupled Device). Digital cinema is independent of how it is projected, meaning that a film that is created digitally and yet displayed using an analog projector is still digital cinema.

One of the pioneers of all-digital filmmaking is George Lucas[10], whose *Star Wars: Episode I – The Phantom Menace* was shot using digital cameras, and only 5% (200 shots) of the film was not altered digitally in some way (Parisi 1999). When the DVD for *Star Wars: Episode II – Attack of the Clones* was released, it was marketed as 'a perfect clone,' since no analog steps were needed between the filming of the movie on digital cameras and its transfer onto DVD. This statement is slightly inaccurate, however, since digital cameras use different compression algorithms to store data than the compression algorithms used on DVDs, but the point is made that there is no 'generational' loss of quality as occurs with analog media – from camera to computer to DVD, the only thing being transferred is computer code.

1)

[10] For further discussion of George Lucas and digital filmmaking, see Barbara Robertson's article on *Star Wars: Episode II – Attack of the Clones* (Robertson 2002).

## *4.2 The Screen*

So far we have only looked at digital cinema from the filmmaker's perspective, but there is another side to the experience, that of the audience. An audience perceives a digital movie (or any movie for that matter) through a screen. Screens could be said to have existed ever since the first humans began painting the walls of their caves. The rock surface became a visual gateway as soon as people began using it to portray the existence of an alternate reality – that of a virtual space. This tradition continued as painting grew into an art form, and as other methods of showing the world through a flat rectangular surface evolved. The Renaissance painter's screen was the canvas, the church's a stained-glass window, and the digital age's a computer monitor.

### *4.2.1 The Dynamic Screen*

Manovich (2001) states that with the invention of cinema, television and video, a new screen emerged: the dynamic screen. While it would actually be more accurate to state that the dynamic screen began with the shadow plays, or shadow theatre, of ancient Asia and Indonesia, Manovich's point is that the dynamic screen shows moving images instead of static pictures, instigating a new relationship between the screen and the viewer. The dynamic screen acts as a window through which the viewer experiences another world – much more so than the static screen – to the point where the ideal dynamic screen is one that is not apparent or noticeable to the viewer at all. People are bothered, for example, when a movie projection does not completely fill the screen, because it causes the screen itself to be visible, thus interrupting the illusion.

The screen is not a neutral medium, but rather an aggressive form that tries to completely immerse a viewer within it (Manovich 2001). Cinema achieves this better than TV or a computer monitor because it has a less obvious frame housing the screen inside of it. Perhaps this will change, however, as large flat-screen technology improves and becomes more prevalent in the household, giving us television screens with minimal borders that take up much less front-to-back space.

> "The wide spread dissemination of home and office flat-screen high-resolution display devices will remediate the presentation of video, and therefore the aesthetics of video production. This technology will become the basis for a new medium, which will relate to the current television in the way that television now relates to film." (Bizzocchi 2001).

### *4.2.2 Beyond the Dynamic Screen*

Lev Manovich sees the dynamic screen as a phenomenon whose era is coming to an end with the introduction of the windowed interface (the computer GUI) and Virtual Reality (VR) technology (Manovich 2001). In VR, for example, the semblance of a screen disappears all together.

To say that the era of the dynamic screen is ending by no means implies that the dynamic screen is dead, however. Just as the static screen is still very much a part of modern society (e.g. painting, photography, graffiti art, *etc*.), the dynamic screen will always have a place, for I doubt that cinema will ever go away. Still, it is interesting to note that new media technologies are ushering in a new relationship between the user and the screen – that where the screen expands to fill a 3-dimensional space in VR, or that of a multi-faceted screen in a windowed interface. With interactive technology the screen also becomes something more: a medium that not only displays, but which can also be interacted with.

> "Interactive virtual worlds, whether accessed through a screen-based or VR interface, are often discussed as the logical successor to cinema and potentially the key cultural form of the twentieth century" (Manovich 2001, p.82).

## ***4.3 The Future of Cinema***

Just as cinema was born from other media arts such as theatre, painting and photography, the future of cinema is the evolution of cinema into another media form. Traditional cinema will remain, but everything that we presently term "expanded cinema," "interactive cinema," "digital cinema," "broadband cinema," "macrocinema," needs its own descriptive name, and one that does not necessarily have cinema in its title.

Manovich (2001) refers to the "next generation of cinema" as a form of digital cinema that adds windows to its language. At present, it is still difficult to get rid of the word *cinema* when discussing future media forms, for we are still entrenched in cinema as a cultural interface. Indeed, in the Object-Oriented Interactive Cinema (OOIC) model I use the term *cinema* in order to place it within a specific cultural and historic context. Some cinema purists may argue that OOIC is not, in fact, cinema, but for certain filmmakers like Peter Greenaway, cinema is more than its medium, encompassing a continuum of moving image technologies. "For Greenaway… Cinema is a continuum. It embraces equally the big movie and the computer screen, the digital image and the handmade film, and – importantly – such structures as speech and writing, acting, editing, light projection and sound" (Rees 1999, p.4). Just as William Gibson coined the term 'cyberspace' in his 1984 novel *Neuromancer*, and it came to define an "on-line world of computer networks" (www.m-w.com), perhaps we need a unique term to describe the new media form that draws on the traditions of cinema but which is a definitively digital construction.

Until that time, I rely on the terms currently being used to discuss future cinematic forms. The next chapter discusses one of these forms: *interactive cinema*.

# Chapter 5:
# Interactive Cinema

Interactive cinema, among other things, puts temporal and spatial control of the movie into the hands of the audience. "As the old boundaries between filmmaker and audience rapidly shift, the new story forms will have far greater complexity and will permit the audience to engage in various ways to shape their cinematic experience" (Beachem 1995). Interactive cinema has great potential for multi-linear story generation, and thus it is worth a look, if only for a historical perspective, at some linear movies which canonically elicit non-linearity.

1. *Timecode* (Directed by Mike Figgis, 2000).
2. *Memento* (Written and directed by Christopher Nolan, 2001).
3. *Waking Life* (Written and directed by Richard Linklater, 2001).
4. *Run Lola Run* (Written and directed by Tom Tykwer, 1999).
5. *Pulp Fiction* (Directed by Quentin Tarantino, 1994).
6. *Rashomon* (Japanese) (Directed by Kurosawa, 1950).
7. Films by Robert Altman (*Nashville* (1975), *The Player* (1992), *Short-cuts* (1993)).
8. Films by Peter Greenaway (*The Falls* (1980), *The Draughtsman's Contract* (1982), *Prospero's Books* (1991)).
9. *Sliding Doors* (Directed by Howitt, 1998).

## 5.1 A New Breed of Cinema

> *"Contemporary storytelling mediums such as the written novel, theater, cinema, and television lack two key capabilities that the computer can offer: autonomy, or the ability to act and change on its own, and interactivity, or the ability to listen, think, and react intelligently to the audience (the user). This ability to collaborate with the user to create stories makes the computer the most powerful medium for interactive fiction."* (Stern in Hirsh 1998, p.16).

There is a clear trend in new media towards interactive systems where images, sound, and even the projection surfaces change in response to conscious and subconscious user activity. This heralds a new breed of cinema – an interactive environment where the one or more movie screens respond to your position, movements, speech, body temperature, *etc*. Interactive cinema offers new ways of approaching user-driven participation media, and forces us to rethink our concept of narrative. Interactive cinema is more than an attempt to redefine an established medium – it is an attempt to create a new way of telling stories.

### 5.1.1 Some Approaches to Interactive Cinema

- **Branching Narrative Interactive Cinema:** Examples include the DVDs *I'm Your Man* (1992) and *Point-Of-View* (2001).
- **Autonomous Agents:** The Interactive Cinema Group (ic.media.mit.edu), directed by Glorianna Davenport, at MIT's media lab, is almost exclusively taking this approach. The premise behind the Interactive Cinema group's work on autonomous agents and automatist storytelling is that the structure of a story can be an emergent property of the interaction of the individual with a storyteller system (Davenport and Murtaugh 1997).
- **Multi-Screen Interactive Cinema:** Central research questions around multi-screen interactive cinema include: How does a multiple screen presentation change the cinematic experience? How does the incorporation of interactivity into

the multiple screen presentation change the cinematic experience? Multi-screen cinema takes away from the privileging of the one screen to the simultaneity of the two or more; the customary authority of the one screen image must now be read spatially as well as temporally.

### *5.2 Some Examples of Interactive Cinema*

- *Point-Of-View* (2001). A branching-type interactive movie on DVD written and directed by David Wheeler. The audience is periodically asked a set of questions while watching the movie, and the responses are used to determine the path through the narrative.
- The German TV project *Morderische Entscheidung* (1991). For the presentation, two different TV stations broadcasted two different versions of the movie at the same time. "One channel followed a woman through the story while on the other channel a man was the hero. The two protagonists met from time to time and it was up to the viewer whom he wanted to follow." (Weilberg 2002).
- *D-Dag*. A millennium project of the Dogma brothers made, in part, as a sequel to *Morderische Entscheidung.*
- *The Last Cowboy* (1998). A media art interactive movie on DVD by Petra Epperlein and Michael Tucker. The story is divided into three parallel video-essays on the myths of America.
- *Lan yue* (1997) by Taiwanese director I-Chen Ko.

  > "I divided the story into five episodes and devoted each 20 minute episode to one set of things [that] happen to this trio. In 2,000 feet, I had to finish every episode. The reels can be screened in any sequence, so you'll have five different endings: either the trio walk down the road holding hands, or they separate, or the woman rejects both men, or she falls for one of the two men. 120 versions, 120 possibilities." (Ko in Weilberg 2002).

- *The Aspen Movie Map* by Andrew Lippman (M.I.T.).

- Jeffrey Shaw's installation *The Legible City.* The participant sits on a stationary bicycle and uses it to navigate through a computer-generated 3-D representation of a city projected on a large screen. In discussing another of Shaw's works called *Point of View* (not to be confused with *Point-Of-View* by David Wheeler listed above) *,* David Rokeby (2002) mentions: "The reward... is the unfolding experience of exploration and discovery, the collection of points of view resulting in a personal reading of the work."
- Myron Krueger, a pioneer in interactive installations, developed a complex set of video-based interactive works called *Videoplace*. The installation uses a video camera to capture the participant, which the computer interprets as a blob or silhouette and uses to generate a response. The responses are designed so that the body becomes a means of creating art, and the interactions are meant to convey the pleasure of aesthetic creation.

### *5.2.1 Timecode*

*Timecode* (2000), directed by Mike Figgis, is four movies in one – the screen is divided up into four quadrants, each one showing a continuous perspective of the overall narrative. On the DVD you can choose which audio stream to listen to while watching the movie, as well as select from other narrative options.

John Purchase (2000) argues that *Timecode* is an object-oriented movie. He compares the traditional narrative film and procedural programming by stating that they are both defined by a sequence of events (Purchase 2000). In *Timecode*, on the other hand, where the movie screen is divided up into four sections, making up four different movie streams, each quadrant can be considered an object. Could the *Timecode* DVD, where the user can manipulate the playback of movie content and audio streams, be an example of Object-Oriented Interactive Cinema?

### *5.3 Computational Narrative*

> *"An intelligent program should be able not only to construct solutions, but also to decide whether or not a given line of action is sufficiently imaginative to present an intrinsic narrative appeal"* (Ryan 1991).

Interactive cinema implies that there is a computational aspect to how the movie is generated for the viewer. The (relatively) easy part is having images appear on the screen and sound heard in response to user input. The hard part is telling the program *which* images and sound to play and *when* to play them based on user input. As discussed in Chapter 3, a random montage of images and sound will probably not produce a very interesting cinematic experience.

> "... the definition of a computational narrative: a narrative whose story representation, structure, and presentation are so intertwined with the functioning of a computational tool that the nature of the narrative reflects the nature of the tool" (Brooks 1996, p.321).

Many researchers have theorized on how a story-generating program should operate. Marie-Laure Ryan discusses storytelling systems from a narratologist's point of view, but her theories are still relevant to interactive cinema, for what is cinema without a compelling narrative? Ryan takes a story-centered approach to narrative generation, believing that an effective program should construct a story world designed to precipitate interesting events, rather than simulating a world and seeing what happens. "If it is true that good plots originate in climaxes rather than in frozen patterns, then story generation should begin with the center(s) of interest, and from there proceed outward, rather than follow a rigidly chronological order" (Ryan 1991).

Ryan's theories are similar to research being done on automatist storytelling agents. Both systems are designed around a large database of knowledge about the narrative universe, including descriptors (stereotyped descriptions of objects), verbs (actions that can be performed by the world's inhabitants), and physics (the physical laws of the environment which govern the objects and characters). "By complying with the rules of its world-knowledge, the program will ensure the logical and pragmatic coherence of the plot" (Ryan 1991).

As an example, Ryan (1991) offers a set of seven natural language instructions that could be used with a story-generating program attached to a large narrative knowledge database:

1. Create a protagonist with a goal.
2. Create an antagonist with a goal contrary to the goal of the protagonist.
3. Create a plan for the protagonist to achieve the goal.
4. Have the antagonist become aware of the protagonist's plan.
5. Have the antagonist develop a plan to counter the protagonist's plan.
6. Have the protagonist proceed along the plan until the scheduled point where the antagonist's plan interferes.
7. Have the protagonist create a new plan to overcome the antagonist and achieve the goal.

Along with these instructions, Ryan (1991) proposes that the program's algorithms should be built so that they:

1. Favor semantic opposition.
2. Favor symmetry and repetition.
3. Try to create suspense.
4. Try to implement functional polyvalence.

I find this abstraction of narrative interesting, and perhaps one day we will reach the level of computational complexity necessary to implement the above instructions and algorithms, but for the moment, I cannot imagine how one would go about creating a story-generating program as Ryan has suggested. The task of creating a large narrative database with snippets of story descriptors, actions, and physical laws, is daunting in itself, but assuming the database existed, the process of constructing compelling stories from these fragments is where the real difficulty would lie. The biggest problem is that the four ‘algorithms’ listed above do not transfer well into computer code. They are far too abstract to be placed into the logical constructs of a programming language. For example, how would one write a computer program that “tries to create suspense,” when the creation of suspense is a subjective process? The algorithm to create suspense would need to be different for every story, and thus the program would need enough intelligence to be able to adapt to unique situations and modify its own code based on past experiences. I do not believe that computer science is quite at the level yet for this to be practical, but perhaps one day.

Nonetheless, many research groups, most notably the Interactive Cinema research cluster at M.I.T., are exploring this type of computational story-generating process, attempting to push computer science, artificial intelligence, and new media to their limits. The next chapter looks at some of these groups.

# Chapter 6:
# Interactive Cinema Research Groups

The purpose of this chapter is to compare some existing examples of interactive cinema with Object-Oriented Interactive Cinema.

## *6.1 The Interactive Cinema Research Group at M.I.T.*

> *"Interactive Cinema reflects the longing of cinema to become something new, something more complex, and something more personal, as if in conversation with an audience"* (M.I.T. Interactive Cinema Group mission statement).

The Interactive Cinema research cluster at M.I.T., currently under the direction of Glorianna Davenport, is asking some interesting questions. Much of their research revolves around how the computer can be used to assist and generate stories, and how the user experience is affected when the cinematic process is made interactive.

> "Though Davenport and her half-dozen graduate students don't suggest an end to the linear narrative technique used in traditional films, they recognize that digital technology has created the possibility of new storytelling techniques that take advantage of variable, non-linear playout of image and sound." (Beachem 1995).

The following are some of the projects produced by researchers in the Interactive Cinema group.

### *6.1.1 Agent Stories*

> "Agent Stories is a story design and presentation environment for non-linear, multiple-point-of-view cinematic stories. It is designed to be placed in the hands of the non-linear story writer to use as a tool to promote structuring and re-writing of non-linear narratives before and as they are realized in audio and video." (Brooks 1996, p.317).

Agent Stories uses a computational model for narrative generation based on splitting the task into three sections:

1. Definition of an abstract narrative structure.

2. Definition of a group of story fragments along with a representation of their associated meaning.
3. Definition of a navigational path through the collection of story fragments.

In agent stories, the computer is represented by a Story Agent, which is an autonomous agent that has the task of reasoning about the story and providing feedback to both the author and participant. "Story Agents allows a story designer to create a simple structure or framework for a story and then use that framework to create multiple narratives from the same collection of story elements" (Brooks 1996, p.322).

*The Structural Environment:*

The structural environment describes the structure of the story in abstract terms. A story framework is "a construction of abstract story element descriptions, referred to as narrative primitives" (Brooks 1996, p.322).

**Agent Stories' Framework (narrative primitives):**

1. Speaker Introduction
2. Character Introduction
3. Conflict
4. Resolution
5. Diversion
6. Ending

Note that this list is similar in concept to Ryan's narrative abstraction in section 5.3, and follows the Aristotelian Arc. In another project by researchers at the Interactive Cinema group, *cinematic* primitives are discussed (Davenport *et al*. 1991). In this project, in order to have practical manifestations for narrative primitives, the researchers suggest that descriptive attributes be attached to shots and sounds as they are recorded by the camera. These descriptions of content could then be used by the computer for more intelligent shot assembly into a final movie.

*The Representational Environment:*

In the representational environment, a story element is presented from a particular point of view, which the authors call a *clip* (Brooks 1996). Clips are linked together through one of the following operations:

1. follows
2. precedes
3. must include
4. supports
5. opposes
6. conflict – resolution

I find it interesting that these words can be seen as computational algorithms or functions built into the narrative program, although the authors do not address this. A *clip* can be seen as a type of object, and thus the framework of Agents Stories touches upon some object-oriented programming concepts. As opposed to branching structures, "Agent Stories offers a different approach, that of storing the story elements in a web or semantic network" (Brooks 1996, p.327). This idea is similar to a property of the Object-Oriented Interactive Cinema model, where multimedia objects are stored in a random-access database.

### *6.1.2 Digital Micromovies*

In a paper entitled "Orchestrating Digital Micromovies," the authors describe how digital technology can be used with a large database of film clips to build narrative structures (Davenport 1993). One of their design considerations was that "a viewer should be able to sit back and enjoy the show without frequent interruptions caused by mandatory interactions" (Davenport 1993). This is an interesting point as many interactive works still employ a 'point-and-click' interactivity which requires input from the user in order to progress though the story.

Similar to other notions of film clip, film object, scene, or shot, the authors define a micromovie as "a short piece of video with descriptive information attached to it which represents a unit of meaning determined by the filmmaker's intent" (Davenport 1993). The most interesting thing about a micromovie is the information that is attached, like metadata, which the program uses to filter video and make decisions. Using keywords to describe the fragments of video, the authors were able to use three different filters in their project – one based on dialog, one based on action, and one based on interaction.

> "The real challenge for interactive filmmakers is to come up with content that will create a compelling and entertaining experience within the framework of an interactive environment" (Davenport 1993).

### *6.1.3 Movies of the Future*

The basic concept behind the "Movies of the Future" project is to "take a computer, infuse it with a detailed database of information about a story, and then let the computer present the story to the audience in its own way" (Beachem 1995). This approach is very much on the generative side of narrative composition, as is similar to Ryan's suggestions (discussed in Section 5.3) of a narrative built from a large knowledge-base.

> "The storytelling computer – responding to the background, interests and preferences of its audience – decides what images or sounds it wants to use in the presentation. It allows the story to take different points of view, choose different characters and scenes, have different pacing and even sets the total running time." (Beachem 1995).

The "Movies of the Future" project deals with a multi-threaded narrative. The authors cite Robert Altman's *Short Cuts* as an example of a traditional film that uses a multi-threaded storyline. Similar to the OOIC model, this model also uses a database of images, sounds and music clips that get assembled into scenes. The clips are also linked, as in Agent Stories (Section 6.1.1), such that one clip can be specified to precede another or be included in the story if another clip is included.

One thing I am unclear on after reading the paper, however, is how exactly the authors are defining a 'clip'. One interpretation is that a clip is just a piece of information, such as a background, a character, or a piece of furniture in the scene, in which case a scene is generated from an elaborate compositing of all the various factors that go into creating an image. The other interpretation is that a clip is an entire pre-recorded image or sound of a particular length. The Object-Oriented Interactive Cinema model attempts to eliminate this kind of ambiguity by stating clear definitions and parameters for its multimedia objects, such as a film clip.

### *6.1.4 Automatist Storyteller Systems*

The Automatist Storyteller Systems project takes an 'editor in software' approach to interactive cinema by attempting to emulate the processes and expertise of the film editor given an indexed database of documentary film segments. "We present a novel approach to documentary storytelling that celebrates electronic narrative as a process in which the author(s), a networked presentation system, and the audience actively collaborate in the co-construction of meaning" (Davenport and Murtaugh 1997).

The system uses keywords connected to story elements to aid the program in selecting relevant fragments and dynamically assembling them into a compelling and coherent narrative. "Connected to the narrative engine through rich feedback loops and intuitively understandable interfaces, the audience becomes an active partner in the shaping and presentation of story" (Davenport and Murtaugh 1997).

## 6.2 Georgia Institute of Technology

### 6.2.1 HyperCafe

Researchers at the Georgia Institute of Technology created an interactive cinema piece called *HyperCafe*, which was an experiment in what they call *hypervideo*. The program placed the user in a type of virtual cafe, where they could explore different scenarios.

> "HyperCafe allows the user to follow different conversations, and offers dynamic opportunities of interaction via temporal, spatio-temporal and textual links to present alternative narratives" (Sawhney et al. 1996, p.1).

In the paper the authors define some terms which approach some of the concepts in the Object-Oriented Interactive Cinema (OOIC) model.

- "Scene: The smallest unit of hypervideo, it consists of a set of digitized video frames, presented sequentially" (Sawhney *et al*. 1996, p.5). This concept of a 'scene' as a distinct unit is similar to the idea of a multimedia object in the OOIC model. In the *HyperCafe* project, however, it was assumed that "video frames carry with them an audio component recorded concurrently with the video frame" (Sawhney *et al*. 1996), whereas in the OOIC model this is not assumed – the video and audio components can be treated as separate objects, one belonging to the class 'video' and the other to the class 'audio'.
- "Narrative sequence: A possible path through a set of linked video scenes, dynamically assembled based on user interaction" (Sawhney *et al*. 1996, p.5). In the OOIC model, the video scenes, which are multimedia objects, are linked through a database of multimedia objects that have defined relationships to each other. The objects are then dynamically assembled based on user interaction. In the *HyperCafe* project user interaction was captured via mouse clicks and buttons; in the OOIC model user interaction is not limited to the standard form of interfacing with the computer – in fact, my installation work uses live video

capture and analysis of the user and their environment (see Chapter 10: OOIC Projects).

- Temporal and spatio-temporal links. Essentially these consist of defined relationships between different scenes based on either a specified time pattern or spatial location. In the OOIC model, these would be seen as operations that can be performed on any of the multimedia objects. The set of operations that can be applied to the multimedia objects make up the rule-set or algorithms that govern the flow and interactivity of the interactive movie. In the *HyperCafe* project, for example, the authors established algorithms that permitted decisions based on (1) random processes; (2) the number of previous visits to that scene; and (3) whether or not the user had previously visited other scenes in the hypervideo space (Sawhney *et al*. 1996).

Other ways in which the *HyperCafe* model treats movie clips as multimedia objects in a database is that, inherent to their construction of scene connections, the narrative sequences could share scenes. Similarly, in the Object-Oriented Interactive Cinema model, a multimedia object can be accessed and played in any order (as defined by the algorithms governing the narrative) and any number of times, even composited on top of itself if the author so desires. Another interesting point about *HyperCafe* is that the authors mention how even though the content of a movie clip remains the same upon each narrative construction, its meaning can change based on it context. For example, a movie clip that is shown near the beginning of the narrative sequence can have significantly different meaning to the same movie clip shown near the end of the narrative sequence. Purposefully giving a movie clip the flexibility to have multiple meanings based on context brings up interesting concepts around composition for a multi-linear narrative.

A last point on *HyperCafe* is that the authors bring up the idea of a user being able to save their narrative sequence “should they wish to return to the program and recover, replay, or rewrite those encounters" (Sawhney *et al*. 1996, p.6). With this ability, users could relive a narrative sequence that they found particularly interesting and enjoyable, although the experience would be changed upon re-viewing since interactivity would no longer be a component of the narrative construction.

### *6.3 Mitsubishi Electric Research Laboratories*

#### *6.3.1 Tired of Giving In*

*Tired of Giving In* uses a storyline based on character development, and "recounts events leading to the Montgomery Bus Boycott, a key moment in the American Civil Rights Movement" (Strohecker 1997, p.377). The characters interact with the participant by filling in gaps in the narrative and adding alternate perspectives through commentary.

The basic structure uses an established story spine, and allows the user to expand into the story's details through interaction. If the user doesn't provide any input, the story continues along its spine, otherwise the characters can be clicked to elicit additional commentary. The system was programmed using an object-oriented language. "Given that so much of the flow and interface cues depend on states of the scenes and characters, we decided that a flexible object-oriented programming environment would best enable implementation of this prototypical structure" (Strohecker 1997, p.378).

# Chapter 7:
# Object-Oriented Interactive Cinema

In discussing Dziga Vertov's *Man with a Movie Camera*, Lev Manovich states:

> *"Vertov is able to achieve something that new media designers and artists still have to learn – how to merge database and narrative into a new form"* (Manovich 2001, p.243).

Object-Oriented Interactive Cinema (OOIC) is an attempt to create a new branch of digital cinema that aims at effectively combining database and narrative. The term *object-oriented* implies a modular system, and OOIC uses the concept of an *object* comparably to object-oriented analysis (OOA) and object-oriented design (OOD) paradigms. Just as in computer programming where an algorithm specifies the steps or functions to be performed on data, the OOIC structure employs algorithms and rule-sets designed to dynamically generate navigation paths from a matrix of interconnected multimedia objects.

> "Hypermedia… refers not only to the manipulation of text objects but to that of images, moving pictures, and sounds; in fact, to any kind of object with unique identity, state and behaviour. Put this way, there seems no alternative to object-oriented programming as a development platform for hypermedia." (Graham 1994).

In computer science, a database is defined as a structured collection of data, or an organized body of related information. There are many different types of databases – hierarchical, network, relational, object-oriented, *etc.* – and they each use different models and algorithms for organizing the data. The Object-Oriented Interactive Cinema model uses an object-oriented database, which stores complex data structures (objects) in hierarchical classes. An object of one class can inherit properties from classes higher than it in the hierarchy.

### *7.1 Object Granularity*

The high-level concept for OOIC is simple – an object-oriented database is composed of multimedia or 'film objects' that are assembled in a multi-linear fashion to create a meaningful narrative. The aim is that, even though the system is using embedded film material, the resultant narrative is a pseudo-emergent property of the participant's interaction with the active cinematic environment. Assembling film objects into a meaningful narrative is a far from trivial task, however. For example, what is a 'film object'? A film object could be a film clip (of any length – from one frame to many minutes of footage), a sound clip (again, of varying length), a collage of multiple film clips, *etc*. This brings up issues of granularity. "Granularity refers to the descriptive 'coarseness' of a meaningful unit of multimedia information" (Davenport *et al*. 1991, p.68).

For example, at what point does a film object become so small as to lose its narrative effectiveness? How does one decide on a suitable level of granularity? Some points can be obvious – for example, a film object of one frame in length would probably have extremely limited use, if any at all, as it would be barely detectable by the human eye. But what about a one second film object? Or a film object an hour long? At what point does a film object become so long that it can no longer be considered a fragment of a modular system? What criteria does one use to determine a suitable level of granularity?

> "Granularity is here defined as the chosen unit size for building story. With it also comes the balance between power and efficiency: by using smaller story granules, there are more ways in which they can fit together, but more work is required for describing how all these pieces can fit together... Conversely, the larger the story granules, the fewer number of ways they can fit together, but the easier it is to put them together... A balance or compromise must be struck, keeping in mind the complexity required by the goal." (Brooks 1996, p.327).

One solution is that the narrative itself could determine what the effective level of granularity should be in a multi-linear film. For example, one narrative might lend itself very well to a music video style of granularity that uses rapid cutting and shot transitions. Relatively short film objects would be most appropriate in this case. Another narrative might require more slower-paced scenes, in which case the narrative may benefit from longer film objects.

In an experiment at M.I.T., researchers used *chunks* and *segments* to refer to lengths of film with some semantic significance; the lengths of these chunks determined the granularity of the system (Davenport *et al*. 1991). In the same experiment the authors referred to a *shot* as "the smallest addressable unit having the finest level of descriptive granularity" (Davenport *et al*. 1991, p.68). A shot consisted of one or more frames, and is analogous to a *video* film object in Object-Oriented Interactive Cinema. In the OOIC model there are three main categories of film objects: video objects, sound objects, and text objects.

In *Hamlet on the Holodeck*, Murray (2000) also uses terms that approach an object-oriented framework. Murray uses the term *frame* to describe story elements: "A frame is a good way for specifying formulaic structures, like a hero and his attributes, or elements of a murder mystery" (Murray 2000). A *frame* is thus analogous to an *object*, so that, for example, the hero becomes an object of the class **Hero**, and inherits all the attributes of that class.

### *7.2 OOIC Composition*

Once the film objects are stored in the database, how is it then decided which objects are shown where, when, and on which screen (i.e. for multi-channel video)? The algorithms that run the decision-making process need to respond to user inputs, allowing the user to construct the story content. One of the biggest obstacles for a user-driven environment, however, is how to create a suitably intelligent interpreter to produce a meaningful narrative out of the assembled film objects based on user inputs.

Theoretically, the algorithms controlling the movie content could come with pre-compiled choices (i.e. pre-made different interpretations of the movie content), essentially giving the viewer a number of pre-made storylines chosen by a professional 'director'. I do not see much increase in viewer satisfaction in this scenario, however, than one would gain from a branching-type multi-linear narrative, and so I have chosen not to include pre-chosen or pre-compiled paths in the OOIC model. To reiterate this another way: the Object-Oriented Interactive Cinema model demands a narrative construction based on user interaction.

### *7.2.1 Embedded versus Emergent Narrative*

> "If we give the interactor complete freedom to improvise, we lose control of the plot. But if we ask the interactor to pick from a menu of things to say, we limit agency and remind them of the fourth wall." (Murray 2000).

In the Object-Oriented Interactive Cinema (OOIC) model, narrative is constructed interactively and in a multi-linear fashion. To produce an effective narrative, the OOIC model tries to find a balance between emergent and embedded story structure. The narrative can be understood as a pseudo-generative path through a database of media objects (video, sound, text), meaning that while the narrative architecture may not be strictly emergent, it has enough variability to appear so. In other words, although there are a quantifiable number of possibilities available to the end-user, the number of possibilities is so large that the narrative appears generative.

### *7.2.2 The Multimedia Database*

> "... creating a work in new media can be understood as the construction of an interface to a database" (Manovich 2001, p.226).

Traversing a multimedia database to construct a multi-linear narrative is one of the fundamental properties of the OOIC model. "An interactive narrative (which can be also called a hypernarrative in analogy with hypertext) can then be understood as the sum of multiple trajectories through a database" (Manovich 2001). An arbitrary path taken through a multimedia database, however, giving an arbitrary sequence of database objects, does not necessarily constitute a narrative. As Ryan argues, "it is simply not possible to construct a coherent story out of every permutation of a set of textual fragments, because fragments are implicitly ordered by relations of logical presupposition, material causality and temporal sequence" (Ryan 2001).

This is why it is so crucial to have effective algorithms that can construct meaningful narratives out of the interaction of the user and the computer. Effectively merging database and narrative will be an important step in the development of compelling interactive digital media, for "a database can support narrative, but there is nothing in the logic of the medium itself that would foster its generation" (Manovich 2001).

> "Given the dominance of the database in computer software and the key role it plays in the computer-based design process, perhaps we can arrive at new kinds of narrative by focusing our attention on how narrative and database can work together" (Manovich 2001, p.237).

Interestingly enough, cinema could already be considered to be at the intersection between database and narrative. "While the audience experiences linear movies as unified entities, filmmakers experience movies as the generation of individual shot and sound elements and the forging of relationships between these elements" (Davenport *et al.* 1991, p.69). During filming, far more footage is captured than will ever be used in the final movie – this raw footage could be seen as a database of film. The editor then uses this database of raw footage to construct a unique trajectory – that which will become the completed movie. We have already seen how different trajectories can be created for the viewer, for movies are often re-released with different versions (a 'Director's Cut,' for example, or a 'cleaned-up' version for television). And when DVD's are released they usually contain a special features section which often includes outtakes, deleted scenes, and footage that did not make it into the theatrical release.

## *7.3 Interactivity in the Context of OOIC*

If we look at interactivity on a continuum, with branching narrative on one end (a fixed number of choices) and generative narrative (autonomous agents) on the other, Object-Oriented Interactive Cinema is in the middle, for although it involves the use of a database of fixed media elements, the sheer number of possible combinations available in even a small matrix gives rise to a pseudo-generative narrative structure. The fact that the film objects can be modified on the fly, using digital filters or effects, rotation or skewing, makes the interactivity in OOIC even more dynamic.

Having said that, however, the Object-Oriented Interactive Cinema model is more of a system or framework for interactive cinema composition than a system based on a particular type of interactivity. This means that using the OOIC model an author could create a work that employs either a very closed or very open interactivity architecture. Section 3.3.5 outlines various categories for interactive works, all the way from external/exploratory (closed) interactivity to internal/emergent transformative (open) interactivity. Using the OOIC model, I believe that an author could create a work in all of these categories. This means that the external/exploratory, internal/exploratory, external/embedded transformative, external/emergent transformative, internal/embedded transformative, and internal/emergent transformative categories are all possible. Because the OOIC model uses film objects of pre-recorded material, however, it is not designed to produce completely emergent narratives from either a first-person or third-person immersive perspective, and thus emergent transformative interactivity is feasible only to a limited extent.

The key concept behind making all of the interactivity options possible is that in the OOIC model, each film object has a set of weighted probabilities attached to it which determine what film object gets played when (typically in response to user input). These sets of probability tables are equivalent to drawing relationships between all of the film objects in the OOIC database. This means that if an author chooses to have a limited number of probabilities leading from each probability node, the movie will be more on the side of branching-type interactivity (embedded transformative interactivity). If, however, the author creates a very complex matrix of film object relationships, with a certain amount of controlled randomness built-in, the piece would be more on the side of generative interactivity (emergent transformative interactivity). The more complex the probability tables, however, the more difficult the task of producing a meaningful interactive narrative. The interactive cinema author must always struggle with this balance.

## *7.4 Conceptual Justification*

As illustrated in Chapter 6, other researchers are also exploring multi-linear interactive cinema from the context of film fragments that are assembled at run-time to construct a narrative. No one has yet, at least in the research that I have performed, attempted to discuss and theorize interactive cinema using an object-oriented paradigm. Having said that, many of the groups have used terminology or concepts that are approaching an object-oriented framework, and so my work fits quite effectively within the larger body of knowledge around interactive cinema by attempting to take existing ideas one step further.

### *7.4.1 Object-Oriented Music Composition*

Chapter 8 on OOIC design discusses how stochastic processes and transition matrices can be used to develop pseudo-generative interactive cinema compositions. While the field of interactive cinema has seen very little research done into these techniques, they are nothing new to computer-aided music design. Musical composers have been using computers to aid in the generation of music since the mid-1950s (Dodge and Jerse 1997), and a good number of musicians have used object-oriented methodologies in their compositional process (Cope 1996, 2000; Krasner 1980; Rodet and Cointe 1984). Lejaren Hiller used computers and transition tables extensively to help generate music beginning in the mid-1950s, and his *Illiac Suite* (1957) is generally considered to be the first computer musical composition (Gena 1994). Computer code can thus be a powerful tool for music synthesis and design, and software programs that generate music can be found using "genetic algorithms, Markov chains, fractal and chaotic algorithms, and various mathematic, connectionist, or pattern-matching algorithms" (Cope 2000, p.15).

The three most relevant concepts to OOIC are how composers have used object-oriented programming, stochastic processes and transition networks to produce music. Aleatoric (random) processes have been used to affect rhythm and pitch[11], to generate fractional noises, and to create chaotic responses in composition (Dodge and Jerse 1997). Conditional probabilities and transition tables have been used for statistical analysis of music, and "afford some possibilities in modeling a randomly generated music on previously composed music" (Dodge and Jerse 1997, p.365).

David Cope has written a number of programs, such as Experiments in Musical Intelligence (EMI), SARA (Simple Analytic Recombinancy Algorithm), and Alice (ALgorithmically Integrated Composing Environment), which use object-oriented techniques in combination with random processes and augmented transition networks (ATNs) to develop new ways of producing music (Cope 1996, 2000). Experiments in Musical Intelligence, for example, uses the Common LISP Object System (CLOS)[12] to compose examples of music in the styles of Bach, Mozart, Chopin, Gershwin, *etc*. using a

1)

[11] For example, Tracy Petersen (1978) used random walks to affect musical parameters over time, such as changing a composition's pitch contour.

[12] CLOS is a powerful object-oriented programming package built into Common Lisp.

database, ATNs, pattern matching, and recombination techniques (Cope 2000). Parallel to Cope's systems, the OOIC model also uses recombinancy of independent entities, which in the case of OOIC are multimedia objects instead of musical bars or notes. And as Cope suggests, for systems that employ recombinancy, "object-orientation is a perfect environment" (Cope 1996, p.125).

OOIC is thus an attempt to apply techniques that have been effective in computer-aided music composition (i.e. object-oriented methodologies, stochastic processes, and transition tables) to the field of interactive cinema – which is essentially computer-aided cinema composition – in the hopes of producing similarly successful and interesting results.

### *7.4.2 A Need for Methodology*

> "Research on multimedia system development shows that contemporary multimedia systems are designed and created primarily by intuition. No methodological support is applied, and projects are characterized by very unsystematic work practices." (Skov and Stage 2002, p.2).

Multimedia systems are often designed without a formal methodology (Pauen *et al.* 1998; Skov and Stage 2002). Frame-addressable videodisks, which emerged in the 1970's, evoked some simple experimentation with branching-type interactivity, where the user could clumsily navigate through the pre-made content stored on the disks. The 1970's and 1980's saw hypertext enter the digital scene, allowing users to navigate text in a non-linear fashion at the click of a mouse button. Single words or phrases suddenly became gateways or branch-points to whole other worlds of information. Unfortunately, hypertext authors quickly became overzealous with this new linking ability, and "what was intended to be encyclopedic became kaleidoscopic" (Davenport and Murtaugh 1997).

Since the invention and rise of new media, digital interactive storytelling has evolved considerably, but it is still a long way from full actualization.

> "Before it becomes a mature art form, several stumbling blocks remain to be conquered, particularly: the need to create a truly systemic approach to narration and story structure; the need to derive a flexible, universally applicable representational schema that describes the form, content, composition, and subtext of media elements; and the need to establish conventions for interaction that are acceptable within the story framework" (Davenport and Murtaugh 1997).

The above quote details the core needs for interactive cinema that the Object-Oriented Interactive Cinema model is trying to address. To summarize in list form, the OOIC model is designed to:

1. Describe a systemic approach to narrative composition and structure, dividing narrative into *story* and *plot*.
2. Present an extensible, universally applicable representational schema, based on an object-oriented class structure, to describe an interactive cinema project's media objects.
3. Present a design methodology for incorporating graph theory, transition matrices, and state diagrams into the composition of multi-linear narratives using film objects.
4. Establish principles for advanced interactivity within a cinematic context by outlining principles for user interaction and navigation.

The OOIC model achieves all of these criteria by applying an object-oriented paradigm to interactive cinema theories.

### *7.5 Object-Oriented Methodologies*

According to some computer scientists, object-oriented concepts and methodologies are relatively new phenomena, having emerged from the historical progression of programming languages. For others, however, object-oriented theories have been around a great deal longer: "The idea that the world could be viewed either in terms of objects or processes was a Greek innovation, and in the seventeenth century, we find Descartes observing that humans naturally apply an object-oriented view to the world” (Booch 1994).

> “It is now commonly held that object-oriented methodologies are more effective for managing the complexity which arises in the design of large and complex software artifacts than either data-oriented or process-oriented methodologies. This is because data and processes are given equal importance. Objects are used to combine data with the procedures that operate on that data. The main advantage of using objects is that they provide both abstraction and encapsulation.” (Preiss, 1998).

The object-oriented (OO) approach consists of breaking down a system into self-contained units, each of which performs a single function. When that function is needed the application *calls* that unit by sending a stream of information through it. These units are called **objects** or **handlers**, and they are connected by a flow of **messages**. “Superficially the term ‘object-oriented’ means that we organize software as a collection of discrete objects that incorporate both data structure and behavior” (Rumbaugh *et al*. 1991).

The concept of an ‘object’ was invented “to manage the complexity of software systems in such a way that objects represented components of a modularly decomposed system or modular units of knowledge representation” (Booch, 1994). Even though each object is considered distinct, it shares the same properties as all other instances of that same type of object. Types of objects are grouped according to particular classifications; in general, “attributes plus optional constraints compose class definitions” (de Champeaux 1993).

The object-oriented domain can be divided into three categories: OO Analysis, OO Design, and OO Programming.

### *7.5.1 Object-Oriented Analysis*

Object-oriented analysis (OOA) takes an object-oriented approach to building models of the real world. OOA involves the definition of system requirements. “Object-oriented analysis is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain" (Booch 1994). The products of OOA serve as the models from which to start an object-oriented design (OOD); the products of OOD can then be used as the building blocks for implementing a system using object-oriented programming (OOP) methods.

The first step in OOA is to create an object model. "Once an object model is available, even a simplified one, the model can be compared against knowledge of the real world or the desired application, criticized, and improved" (Rumbaugh *et al*. 1991). Under OOA falls the responsibility of defining a notation for depicting object and class models for the system under design. The process of object definition is crucial in OO development because it enables reusability. "Well designed objects in object-oriented systems are the basis for systems to be assembled largely from reusable modules, leading to higher productivity" (Graham 1994).

### *7.5.2 Object-Oriented Design*

Object-oriented design (OOD) methods "emphasize the proper and effective structuring of a complex system" (Booch, 1994), as opposed to programming methods which are focused on the effective implementation of language mechanisms. The design phase is where class and object relationships are defined. Under the Object-Oriented Interactive Cinema model, the design stage is characterized by specifications for multi-linear narrative generation using OO concepts and graph theory.

### *7.5.3 Object-Oriented Programming*

> "Object-oriented programming is a method of implementation in which programs are organized as cooperative collections of objects, each of which represents an instance of some class, and whose classes are all members of a hierarchy of classes united via inheritance relationships" (Booch 1994).

Object-oriented programming (OOP) is implemented using an OOP language, such as C++, Java, SmallTalk 80, CLOS, Eiffel, *etc*. Object-oriented programming languages are favored for many software development projects because they make systems “more flexible, more easily extensible and less costly to maintain” (Graham 1994).

### *7.5.4 OOIC Implementation*

The Object-Oriented Interactive Cinema model describes a framework for OOA and OOD for interactive cinema, but stops there. OOIC does not define specific implementations in an OOP language, but rather keeps the conceptualization more abstract, leaving the coding open for programmers to design an OOIC system according to their own language preferences.

As a model, OOIC omits nonessential details through abstraction, which helps the interactive cinema creator deal with the system’s complexity. “A model is an abstraction of something for the purpose of understanding it before building it... Abstraction is a fundamental human capability that permits us to deal with complexity” (Rumbaugh *et al*. 1991). The OOIC model is aimed at aiding the interactive cinema developer by providing precise notations and procedures so that details can later be added to transform the design into an implementation.

### *7.5.5 The Process of Object-Oriented Development*

The steps for object-oriented development can be broken down into macro and micro processes. The OOIC model is developed using these procedures.

**Macro process:**

1. Establishment of core requirements (conceptualization). The core requirements have been outlined in previous sections of this thesis, especially in Section 7.4.1: Conceptual Justification.
2. Development of class and object models for the system (analysis). The OOA phase is described in Section 7.6.

3. Creation of a design architecture (design). The OOD phase is described in chapter 8.
4. Implementation (programming). The OOP phase is outside of the scope of this thesis.
5. System management (maintenance). The maintenance phase is outside of the scope of this thesis.

**Micro Process:**

1. Identification of the classes and objects (abstraction). This is handled in the OOA phase.
2. Identification of the relationships among these classes and objects. This is handled in the OOD phase.
3. Specification of the interface and implementation of these classes and objects. This is handled in the OOP phase and is outside the scope of this thesis.

## *7.6 Object-Oriented Analysis*

"... object-oriented development provides the tools, such as abstraction, encapsulation, and inheritance, to build libraries of reusable components" (Rumbaugh *et al*. 1991). Two fundamental concepts for object-oriented analysis are abstraction and encapsulation.

### *7.6.1 Abstraction*

An abstraction:

- Is a simplified description or specification.
- Is a description that emphasizes some of the system's details (those that are significant to the user) while suppressing others (those that are, at least for the moment, less significant).

- Serves to separate an object's essential behavior from its implementation or accidental properties.

Grouping objects into classes is a way of abstracting a problem. "An abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of objects and thus provides crisply defined conceptual boundaries, relative to the perspective of the viewer" (Booch 1994).

### *7.6.2 Encapsulation*

Encapsulation is a form of information hiding, where all the non-essential characteristics of an object are hidden. Through encapsulation an object's methods are visible and accessible, but the implementations for these methods are hidden inside the object. "Abstraction and encapsulation are complementary concepts: abstraction focuses upon the observable behavior of an object, whereas encapsulation focuses upon the implementation that gives rise to this behavior" (Booch 1994).

### *7.6.3 Identity*

There is some dispute amongst computer scientists about what exactly constitutes an object-oriented approach (Rumbaugh et *al*. 1991), but there are a number of concepts which are widely held to be essential properties. Identity is one of them.

"Identity means that data is quantized into discrete, distinguishable entities called objects" (Rumbaugh et *al*. 1991). A wheel of a bicycle, a paragraph in a document, and a black pawn in chess are all examples of objects. Every object is distinct, even if it has shared attribute values with other objects. An object is a unit of structural and behavioral modularity that has properties (Buhr 1988), although an object's identity means that it is distinguishable beyond its descriptive properties. Besides an identity, an object has a state and behavior(s); "the structure and behavior of similar objects are defined in their class"

(Booch 1994). We use the notion of an *instance*, which is conceptually interchangeable with object, when we want to emphasize that an object is a 'member' of a certain class.

"Objects serve two purposes: They promote understanding of the real world and provide a practical basis for computer implementation" (Rumbaugh *et al*. 1991). At a very young age, humans learn the *object concept*, which is that objects, such as a red ball, have permanence and identity apart from any operations that are done to them, so that we know that an object still exists even after leaving our field of view (Booch 1994). This is to say that although object-oriented is typically associated with computer science, the object concept is a property of human cognition. In this light, an object can be a tangible thing, an intellectual thing, or "something toward which thought or action is directed" (Booch 1994).

**Notation:**

The notation used for OOIC is a hybrid of the OMT (Object-Modeling Technique) notation used by Rumbaugh *et al*. (1991) and OMG's UML (Unified Modeling Language) (www.omg.org/uml/).

An object instance is depicted as a rectangle with rounded corners containing the object name. Typically the object's associated class is also written inside the rectangle, above the object name, in parentheses and in bold. In Figure 1 below, the instance is named 'Clip_1' and it belongs to the class 'VideoClip'.


**Figure 1:** An instance diagram

*7.6.4 Classification*

"Classification means that objects with the same data structure (attributes) and behavior (operations) are grouped into a class" (Rumbaugh *et al*. 1991). "A class stands for a family of objects that have something in common" (de Champeau *et al*. 1993). **BikeWheel**, **Paragraph**, and **ChessPiece** are examples of classes. Like an object, a class is an abstraction, and defining classes for a system is arbitrary, but highly dependent on the needs and goals of the application. Part of the OOA phase is the definition of the class structures for the system. As already mentioned, an object is an instance of a particular class. For example, a black pawn is an instance of the class **ChessPiece**; a Porsche is an instance of the class **SportsCar**; a bicycle object, of the class **Bike**, may contain two wheel instances of the class **BikeWheel**.

**Notation:**

A class is represented in OOIC notation as a rectangle with the name of class written inside in bold.


**Figure 2:** A class diagram

In summary so far:

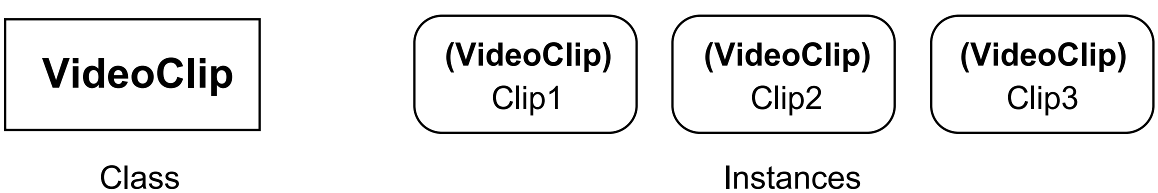


**Figure 3:** A class with three object instances

### 7.6.5 Attributes

Classes are described by their *features*, which is a generic word to describe either attributes or operations. “Attributes are seen as special binary relations that help define a prototypical instance of a class” (de Champeaux 1993). Car make, for example, may be seen as a binary relation between the class of **SportsCar** and the enumerated domain *{Porsche, Ferrari, Lamborghini, Aston Martin, Maserati}*. An attribute is a data value held by the objects of a class. Each object instance has its own unique attribute value, even though the attribute name is shared between all objects of the same class.

**Notation:**

Attributes are listed in the class rectangle underneath the class name, separated by a horizontal line. A horizontal line is not used in instance diagrams. Attributes may be followed by their data type (e.g. string, integer, float, Boolean, *etc*.). An attribute can also have a default value.

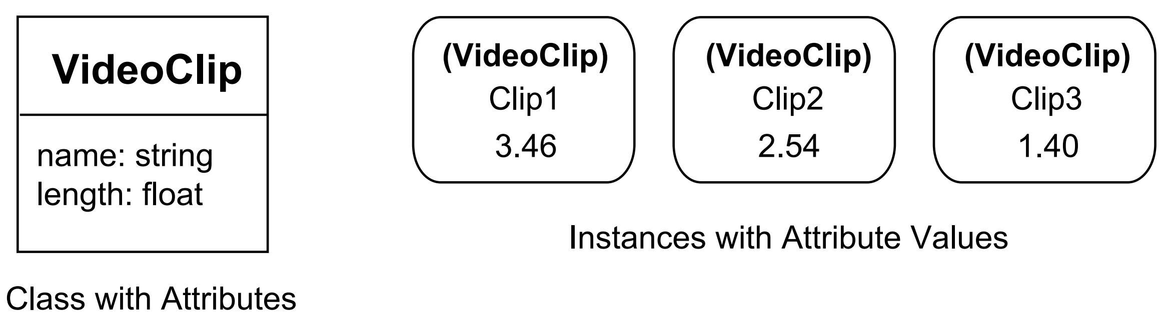


**Figure 4:** Class and instance diagrams with attributes

### 7.6.6 Operations

“An operation is a function or transformation that may be applied to or by objects in a class” (Rumbaugh *et al*. 1991). Operations can be polymorphic, meaning that the same operation takes on different forms in different classes. The transportUp() operation, for example, may have a different implementation in the **Elevator** class than in the

**Escalator** class, even though the operation has the same effect in each class of moving people vertically upwards. A *method* is the implementation of an operation for a class.

**Notation:**

Operations are listed in the class diagram underneath the attributes, separated by another horizontal line. Values that can be passed to the operation, called arguments, are listed in parentheses after the operation name. An operation that does not require any arguments is followed by opening and closing parentheses, as in the transportUp() operation mentioned above.

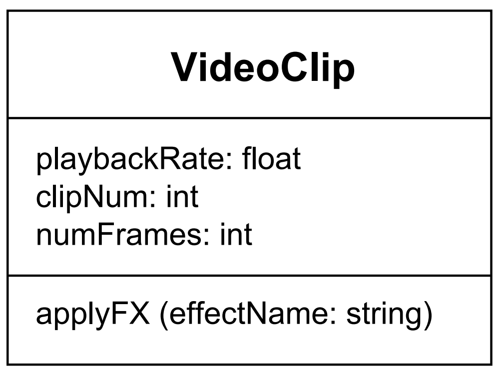


**Figure 5:** A class diagram with attributes and operations

In the above example, the class **VideoClip** has an operation applyFX() with an argument *effectName* of type *string.*

In summary:

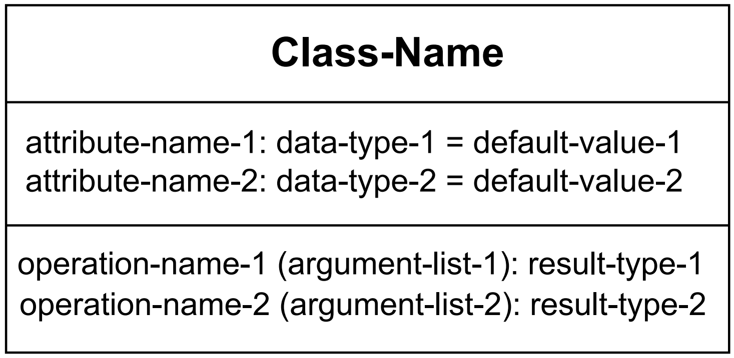


**Figure 6:** A generic class diagram

### *7.6.7 Associations*

An association characterizes a group of links with common structure and common semantics. It describes a set of potential links in the same way that a class describes a set of potential objects; a link is a physical or conceptual connection between object instances (Rumbaugh *et al*. 1991). A link is an instance of an association.

**Notation:**

Links are drawn as straight lines connecting classes or instances. A descriptive word is written next to the line to describe the nature of the association. Links are typically directional and are read from left to right, although arrows can be used for increased clarity.

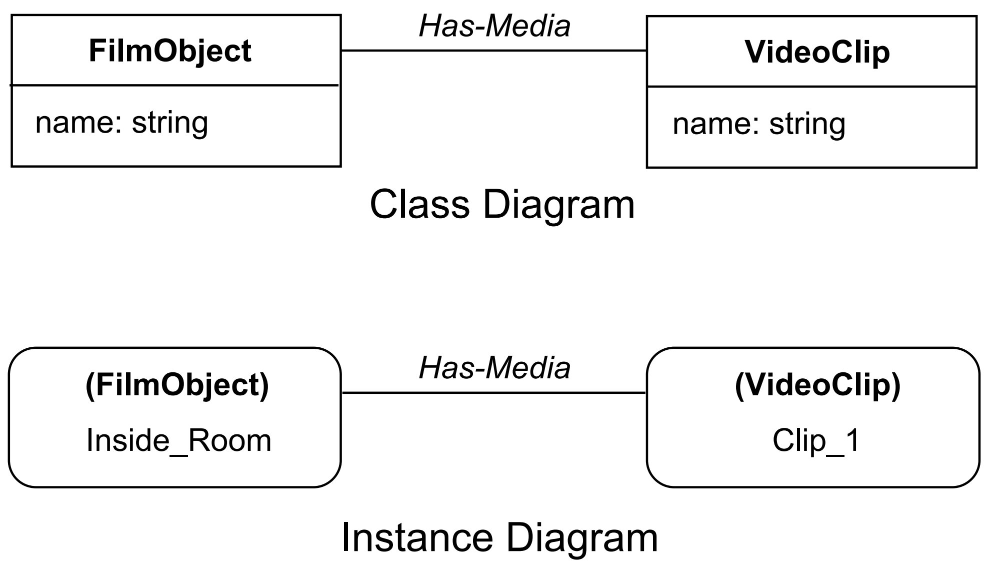


**Figure 7:** Associations between classes and objects

### *7.6.8 Inheritance*

Inheritance is a product of generalization. Generalization is when a class, being the *superclass*, is refined into *subclasses*. "Attributes and operations common to a group of subclasses are attached to the superclass and shared by each subclass" (Rumbaugh *et al*. 1991). A subclass is said to *inherit* the features of its superclass. Each instance of a subclass is also an instance of the superclass as well, except that the subclass adds its own specific attributes and operations. In this way, the features of the superclass do not need to be repeated in each subclass – the features are automatically inherited and available.

Inheritance is a fundamental concept of the object-oriented paradigm. Inheritance involves hierarchical abstraction. "Hierarchical abstraction of common features contributes to the overall human understanding of the objects and classes comprising a system" (de Champeau 1993). Often all classes within a system are considered to be subclasses of a common base class, which serves as the root of the inheritance hierarchy and defines no attributes, operations, or transitions.

**Notation:**

In OOIC notation, inheritance is drawn as lines between the superclass and its subclasses, with a triangle underneath the superclass. In Figure 8 below, **Video**, **Text**, and **AudioClip** are all subclasses of the **Media** superclass, and thus they each inherit the *name, startTime* and *endTime* attributes.

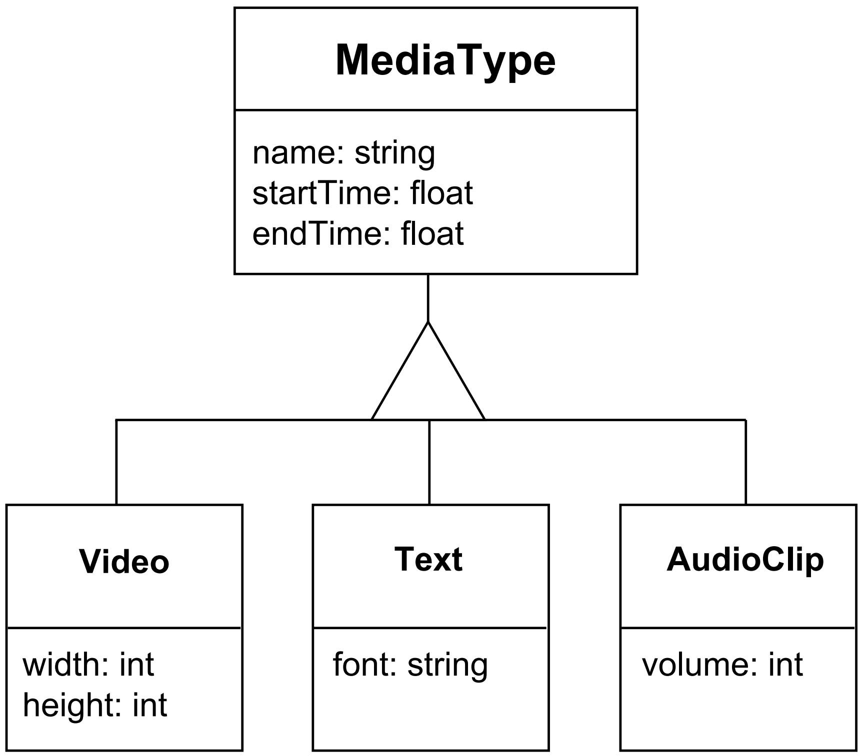


**Figure 8:** A class diagram with inheritance

*7.6.9 Multiplicity and Aggregation*

Multiplicity specifies the number of instances of one class that may relate to a single instance of an associated class (Rumbaugh *et al*. 1991). Aggregation is where “objects representing the components of something are associated with an object representing the entire assembly” (Rumbaugh *et al*. 1991). For example, a document is an aggregation of paragraphs, and a paragraph is an aggregation of sentences.

**Notation:**


**Figure 9:** Multiplicity and aggregation

In Figure 9, the 1+, 10+ and solid dots in the association links are multiplicity symbols. The solid dot means that there are zero or more examples of how instances of one class may relate to a single instance of an associated class. The 1+ means that one or more *VideoClip* objects are needed to form a *Movie* object. A *VideoClip* object is a type of scene, and thus it makes sense that you cannot have a movie without at least one scene. Likewise, the 10+ means that at least 10 frames are needed for a video clip. The number 10 is arbitrary, but is probably close to the minimum number of frames that can be detected by the human eye as a meaningful scene (at a playback speed of 15 frames per second or greater). In diagrams where multiplicity symbols are not used, the implication is that there is one and only one association between the classes.

A diamond is the symbol for aggregation. In Figure 9 the diamonds tells us that a *Movie* object is an aggregation of *VideoClip* objects, and that a *VideoClip* object is an aggregation of *Frame* objects.

## *7.7 OOIC Class Structures*

Continuing with the object-oriented analysis phase, this section outlines the class structures for OOIC.

### *7.7.1 The **InteractiveMovie** Class*

The final product in an OOIC project that is presented to the viewer is an interactive movie. Figure 10 illustrates the two most important classes in the Object-Oriented Interactive Cinema model: the **InteractiveMovie** and **FilmObject** classes.

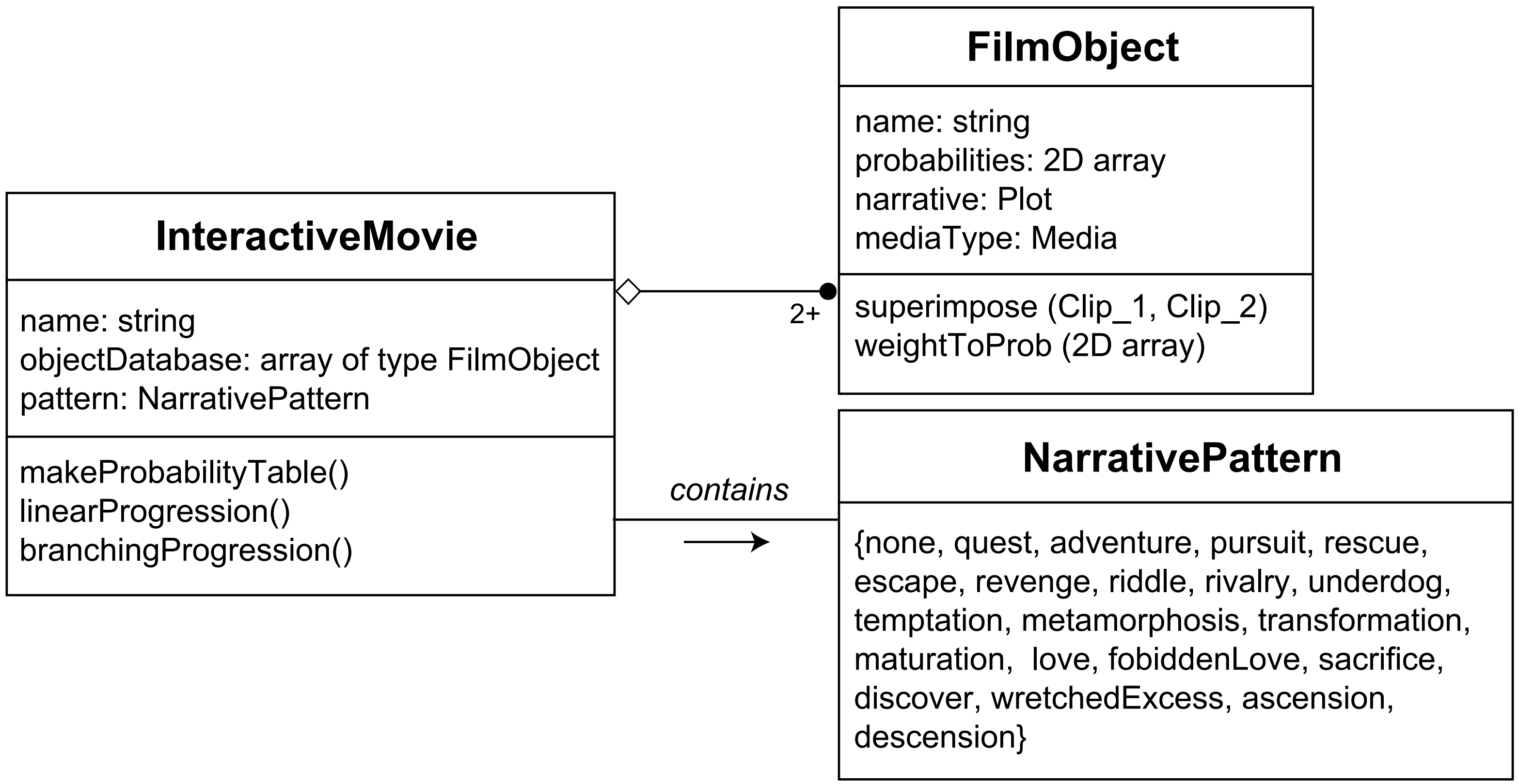


**Figure 10:** The **InteractiveMovie** class with associations

An *InteractiveMovie* object is an instantiation[13] of the **InteractiveMovie** class; it is a working example of interactive cinema. As can be seen in Figure 10, the **InteractiveMovie** class has three attributes: *name* of type *string*, *objectDatabase*, which is an array of type *FilmObject* that is composed of instances of the **FilmObject** class, and *pattern* of type **NarrativePattern**. Three operations are also listed for this class; these are discussed in the next chapter on object-oriented design.

The **FilmObject** class is described in the next section.

The association between the **InteractiveMovie** and **FilmObject** classes indicates that an *InteractiveMovie* object is an aggregation of *FilmObject* objects, and that two or more instances of the **FilmObject** class are needed to make an *InteractiveMovie* instance. Obviously, to make a compelling interactive movie, many more than two film objects would be required.

**NarrativePattern** is a user-defined enumerated type[14] which describes the theme of the narrative pattern, if any. As an example, in Figure 10 I am using the twenty 'master' narrative patterns as proposed by Tobias (1993) and discussed in Section 3.1.1.

### 7.7.2 The ***FilmObject*** Class

The foundational principle of the OOIC model is that an object-oriented database, composed of film objects, is used to assemble a multi-linear cinematic narrative. Thus, the film object is the fundamental building block for an interactive movie.

1)————————————

[13] Instantiation refers to producing a particular object from its class template. This involves initialization of instance variables with either default values or those provided by the class instantiation function (the constructor). An instantiation is the representation of an abstraction (a class) by a concrete example (an object).

[14] An enumerated type is "a type which includes in its definition an exhaustive list of possible values for variables of that type" (dictionary.reference.com). Although an enumerated type is a data type, it can also be seen as a special kind of class, and thus enumerated types are represented using class notation.

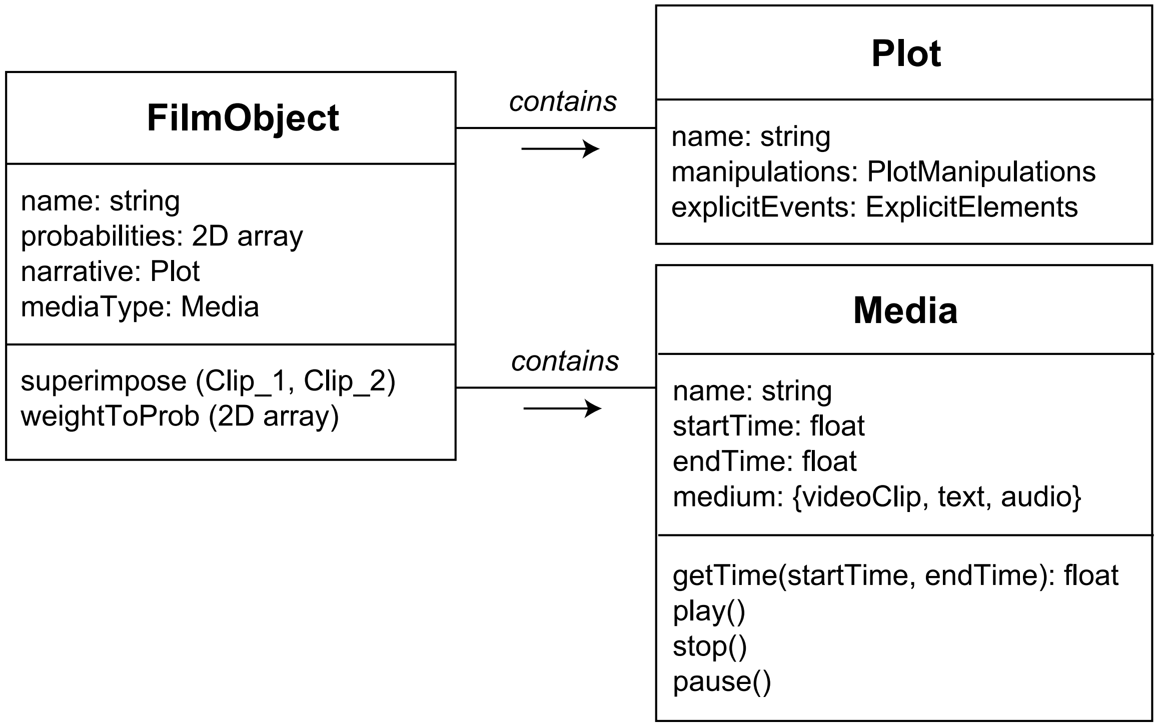


**Figure 11:** The **FilmObject** class with links to the **Plot** and **Media** classes

The **FilmObject** class has four attributes: (1) *name*; (2) *probabilities*, which is a 2-dimensional array of type *float* (a 2-dimensional array can be envisioned as an array of arrays), and refers to transition probability values (discussed in the next chapter); (3) *narrative* of type **Plot**; and (4) *mediaType* of type **Media**. The attributes *narrative* and *mediaType* indicate that a film object is composed of a narrative element and is communicated through a particular medium, representing the object's content and form, respectively.

The operation superimpose(Clip_1, Clip_2) is discussed in Section 7.8.3, and the operation weightToProb(2D array) is discussed in Section 8.4.2.

### 7.7.3 *The **Plot** Class*

*Plot* objects represent the building blocks for narrative content in the interactive movie. A plot was defined in Section 3.1.4 as being composed of the explicitly presented narrative events as well as the narrative's *nondiegetic* material. Because the OOIC class structure is a model for everything visibly and audibly present in an interactive movie (i.e. the plot), the story elements that a user *infers* from the plot are not included within the OOIC model.

The **Plot** class has three attributes: *name*, *manipulations*, and *explicitEvents*. The *explicitEvents* attribute represents the elements of the plot that overlap with the story, and is of type **ExplicitElements** (discussed in the next section).

The *manipulations* attribute is of type **PlotManipulations**, and refers to a meta-tagging class which is used to help identify and describe the techniques contained in the film object that manipulate the plot. The **PlotManipulations** and **ExplicitElements** classes are both meta-tagging classes, and are used to attach meaningful descriptors to film objects as a means of identification, organization, and/or algorithmic manipulation.

The *perspective* attribute of the **PlotManipulations** class is of enumerated type *{internal, external}*, and refers to the internal/external types of interactivity categories discussed in Section 3.3.4. *Internal* essentially indicates that the film object is presented from the first-person perspective, or that the user is directly part of the narrative and represented by a story character, with *external* indicating a third-person perspective, or that the user is external to the narrative.

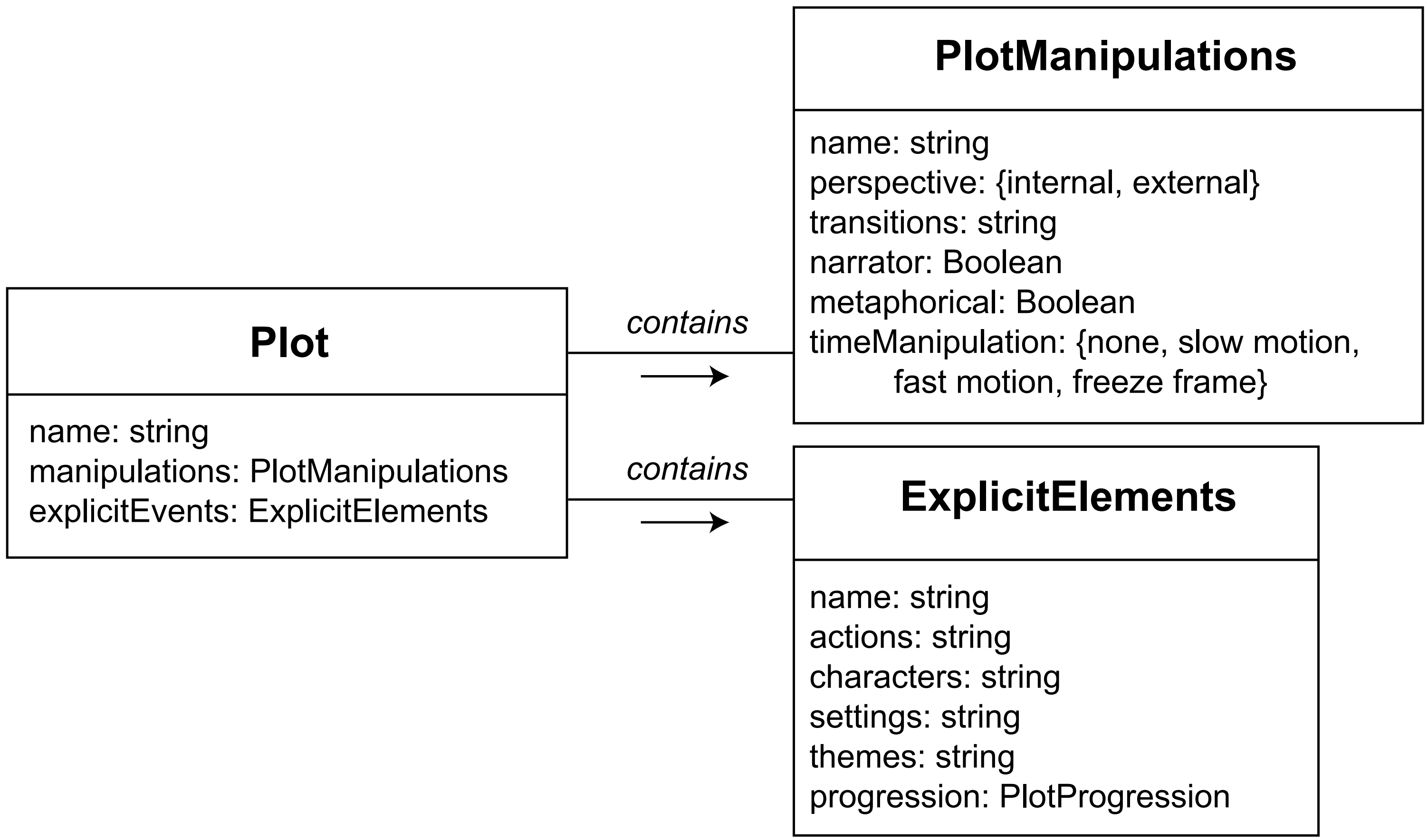


**Figure 12:** The **Plot** class with associations

The *transitions* attribute of the **PlotManipulations** class is meta-tagging attribute, and can be used to organize or manipulate a film object based on the cinematic transitions it uses (e.g. cross-dissolve, cut, wipe, *etc*.). The *narrator* attribute is of type *Boolean*, indicating that it has the value of either true (or any non-zero integer) or false (or zero). This attribute essentially asks the question of whether the particular film object has a narrator's voice as part of its discourse. The metaphorical attribute is also of type Boolean, and asks the question of whether the film object's content is metaphorical. The *timeManipulation* attribute is of enumerated type *{none, slow motion, fast motion, freeze frame}*, and, as the name implies, refers to how time is manipulated in the film object (if at all).

### *7.7.4 The **ExplicitElements** Class*

As discussed in Section 3.1.1, a narrative is often divided into *story* and *plot*. Because the *entire* story is something created and inferred by the reader, it cannot be fully represented in such a discrete form as a class diagram. Plot, however, being something more quantifiable as what is presented to the viewer, is suitable as a class construction. Since story and plot overlap in that they both share the explicitly presented events of the narrative, the **ExplicitElements** class is used to represent those aspects of the plot which are shared by the story.

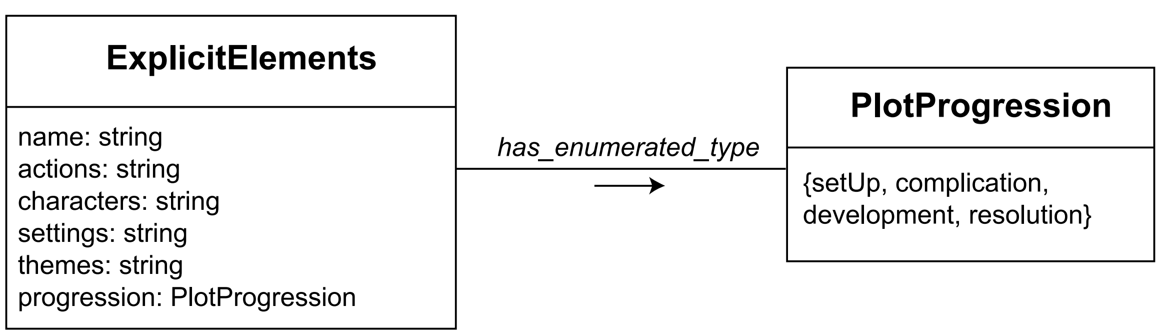


**Figure 13:** The **ExplicitElements** class with **PlotProgression** association

The **ExplicitElements** class has five attributes of type *string*: *name*, *actions*, *characters*, *settings*, and *themes*. These are used to describe a particular object's explicitly presented elements and distinguish it from the explicitly presented events of other film objects. The **ExplicitElements** class is a meta-tagging class, and is used to attach meaningful descriptors to film objects as a means of identification, organization, and algorithmic manipulation. For example, the *theme* attribute could be used similarly to Murray's use of *theme* to refer to "a generic narrative unit that can be fit into multiple narratives, a unit such as the departure of a hero, the catalog of ships, the dressing of a hero for battle, the boast of a hero before battle, and the death of a hero" (Murray 2000).

The **ExplicitElements** class also has the attribute *progression*, which, as the name implies, refers to how an *ExplicitElements* instance fits in to the rest of the narrative's overall flow. The *progression* attribute is of type *PlotProgression*, which is a user-defined enumerated data type. In Figure 13, I have used the Aristotelian Arc categories *{setUp, complication, development, resolution}* as the values for the **PlotProgression** enumeration, but these can be any user-defined values; it is up to the interactive cinema author to decide if they want to employ a traditional narrative arc or use unique values outside of standard constructions (see Section 3.2.2 on multiform narrative for further discussion).

If the interactive cinema author wishes to establish cause and effect relationships in the narrative, the *progression* attribute can achieve this. For example, if, as in Figure 13, the Aristotelian Arc categories were used for the **PlotProgression** enumerated type, the author could establish the following algorithm (written here in the programming language *Python*):

```
if currentClip().progression == 'SetUp':
     playComplicationObject()
elif currentClip().progression == 'Complication':
     playDevelopmentObject()
elif currentClip().progression == 'Development':
     playResolutionObject()
else:
     playSetUpObject()
```

The above example uses dot notation to refer to the *progression* attribute of the currently playing film object. Dot notation is commonly used in object-oriented programming languages, and provides a shortcut for invoking methods and for setting or querying attribute values. The syntax for dot notation is as follows:

```
object.attribute
```

or

```
object.method(<arguments>)
```

In the example, the `currentClip()` function is used to return the name of the currently playing film object, and then dot notation is used to access the object's *progression* attribute. A series of comparisons are then made to detect which of the four Aristotelian Arc categories the current film object belongs to (`elif` stands for 'else if'), and an operation is called accordingly. For example, if the currently playing film object had the value of 'SetUp' for its *progression* attribute, the `playComplicationObject()` function would be called, causing a film object with a 'Complication' *progression* attribute to be the next film object played.

### *7.7.5 The **Media** Class*

The **FilmObject** class has the attribute *mediaType* which is of type *Media*.

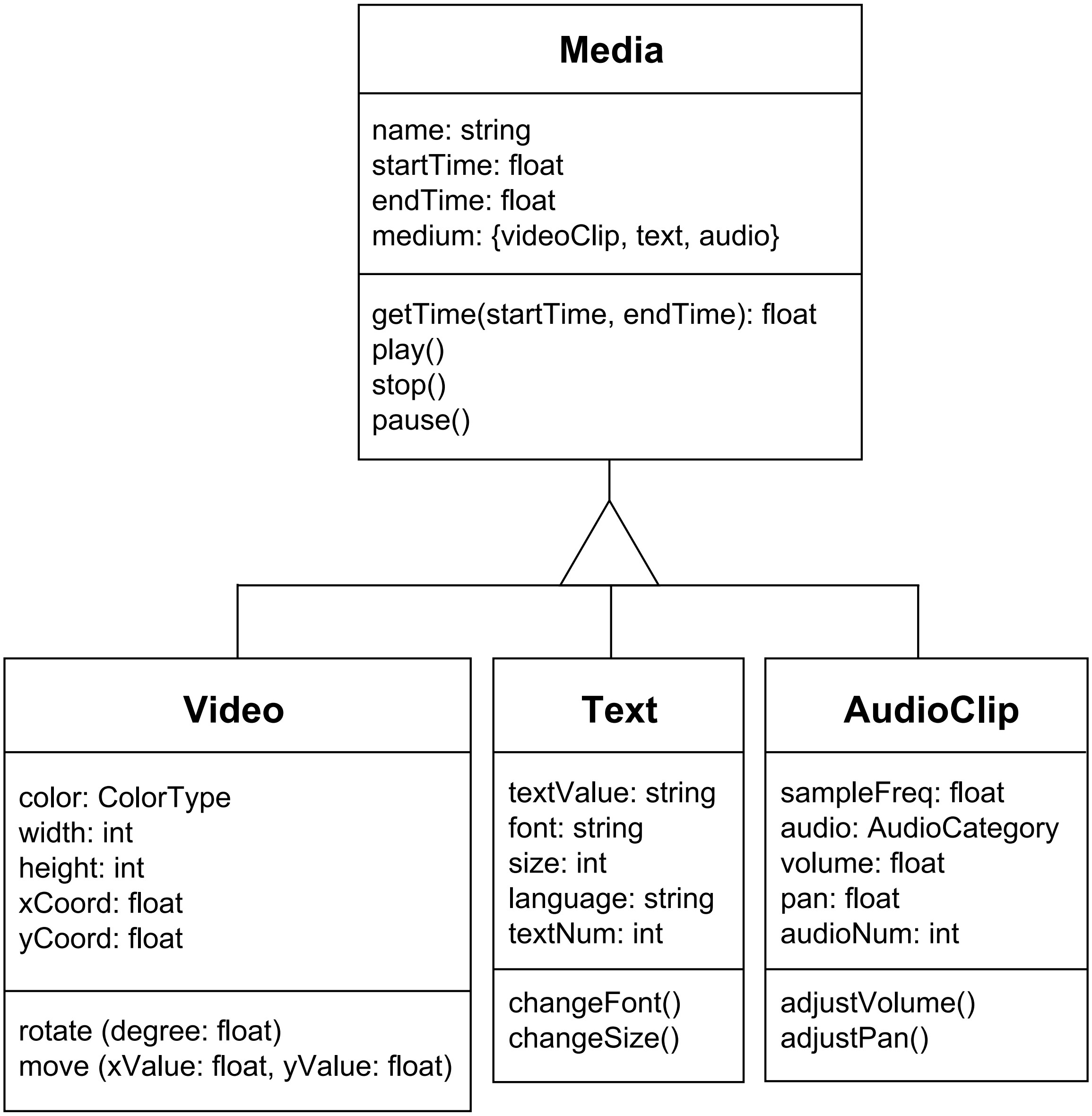


**Figure 14:** The **Media** class with subclasses

The **Media** class contains all the non-narrative elements of a film object – essentially, the media forms. It contains the subclasses **Video**, **Text**, and **AudioClip**, which are the only types of media available to a film object under the OOIC model (although, theoretically, an author could create additional media classes as required). The three subclasses inherit all of the attributes and operations of the **Media** class.

**Media** has three self-explanatory attributes: *name*, *startTime* and *endTime*. The two time attributes are floating-point numbers and can represent any unit of measurement the author so desires (e.g. minutes, seconds, milliseconds). A film object can have a video clip, text, or an audio clip as its media component, and thus the *medium* attribute, which is of the enumerated type *{videoClip, text, audio}*, can be used with conditional algorithms to access attributes and/or functions specific to the object's media type.

The *startTime* and *endTime* attributes are used in the getTime() function to calculate the total running time of the media object. The operations play(), stop(), and pause() are playback controls.

The **Video** and **AudioClip** classes are discussed in sections 7.7.6 and 7.7.7 respectively.

The **Text** class describes properties for instances of on-screen text that can be used in a movie. *Text* instances can be used for the opening credits, end credits, subtitles, and for any other effect where on-screen words are desired. The **Text** class has five attributes: (1) *textValue*: this variable stores the actual words, spaces, and punctuation of the text; (2) *font*: this represents the text's font, such as Times New Roman, Verdana, or Courier; (3) *size*: this is the point size of the text (e.g. 10, 12, 14); (4) *language*: this is the text's language, such as English (U.S.), English (U.K.), French, Spanish; and (5) *textNum*: this is a cataloguing or ordering number which can be used for organization during playback. The two operations in the **Text** class enable the font and/or size of the text to be changed.

### *7.7.6 The **Video** Class*

The **Video** class and its subclasses describe the graphical media components of an interactive movie. Using the term *video* in reference to cinema could be problematic for some cinema theorists, as traditionally video has been associated with television or videotape technology, but I am using the word here with the assumption that digital cinema is a form of video since celluloid film is not used. Watching *Casablanca* on videotape in a VCR is still a cinematic experience, of course, and so I am defining a movie as something more than the format on which it is recorded or presented back to the viewer.

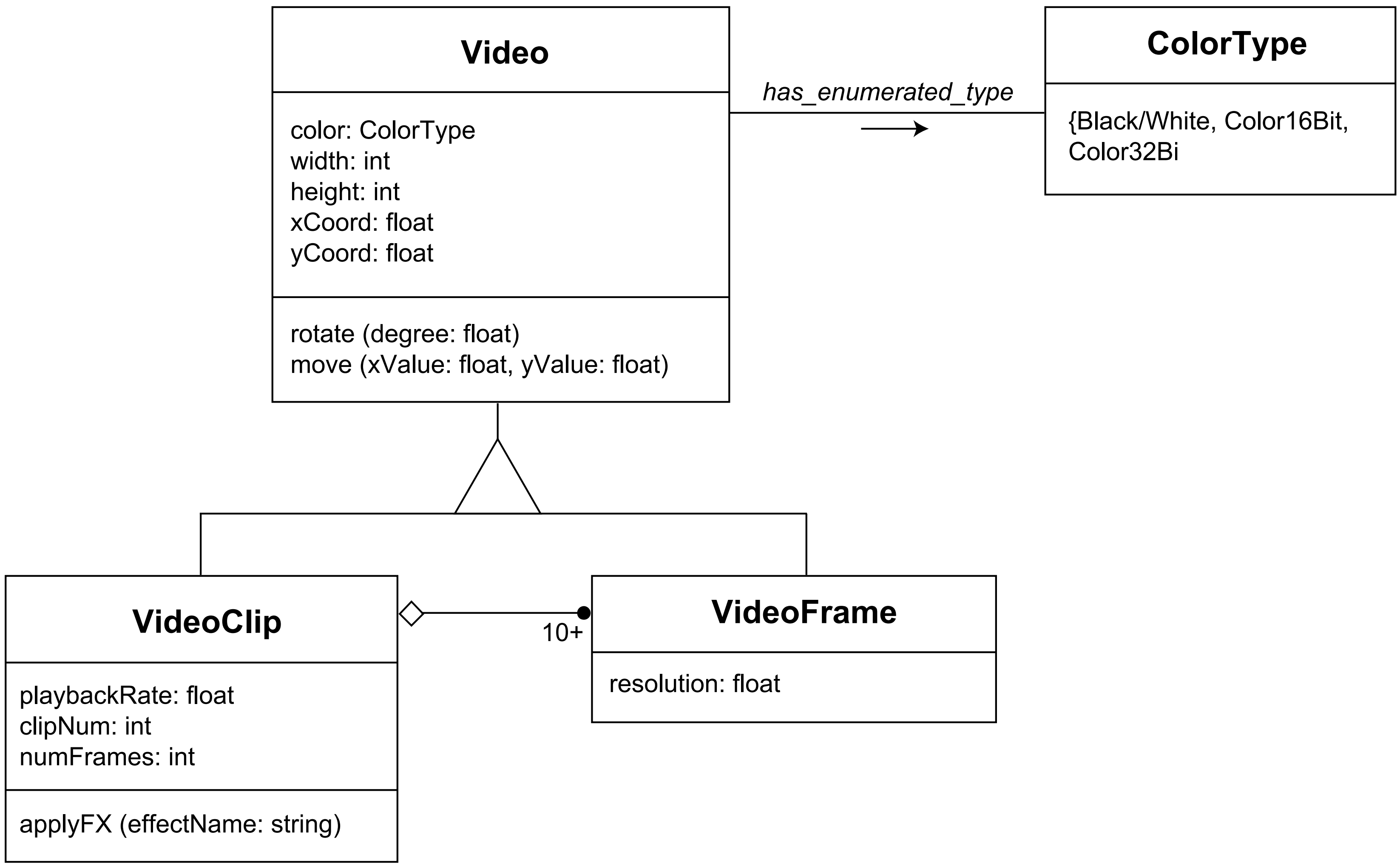


**Figure 14:** The **Video** class with subclasses and link to **ColorType**

The **Video** class has five attributes which are also inherited by its subclasses: *color* of type **ColorType** (an enumerated type), and four attributes which describe the on-screen characteristics of the video: *width*, *height*, *xCoord*, and *yCoord*. The operations serve to manipulate the attributes by to changing the x, y coordinates of the video – the move() function – or by rotating the video around its center point – the rotate() function.

The **Video** class has two subclasses, **VideoClip** and **VideoFrame**. A *VideoClip* instance is made up of an aggregation of ten or more *VideoFrame* instances. A *VideoFrame* object is a single frame of video, with its only unique attribute being its resolution in pixels (e.g. 2 megapixels, or 2,000,000.0). Being a subclass of **Video**, which is itself a subclass of **MediaType**, **VideoFrame** inherits all of the **Video** and **MediaType** class attributes, although the *startTime* and *endTime* attributes for a *VideoFrame* object would be identical.

The **VideoClip** class is a crucial component of the OOIC model. Video clips are analogous to movie 'scenes' – they are the visual components of *FilmObject* instances which get assembled, based on user input, to form the visual content of the interactive movie. In addition to the inherited attributes, the class has three unique attributes: *playbackRate*, which is the playback speed of the movie (in frames-per-second); *clipNum*, which is a cataloguing number for the *VideoClip* instance and a way of ordering the clips for playback; and *numFrames*, which refers to how many frames are contained in the video clip object.

The most interesting operation unique to the **VideoClip** class is the applyFX() function. This operation can be used to apply visual effects to the video clip in real-time, such as solarize, distort, add noise, emboss, black & white, slow speed, fast speed, color saturation, find edges, *etc*. The applyFX() operation involves the real-time rendering of special effects applied to the original video footage.

### 7.7.7 *The **AudioClip** Class*

The final component of **MediaType** is the **AudioClip** class.

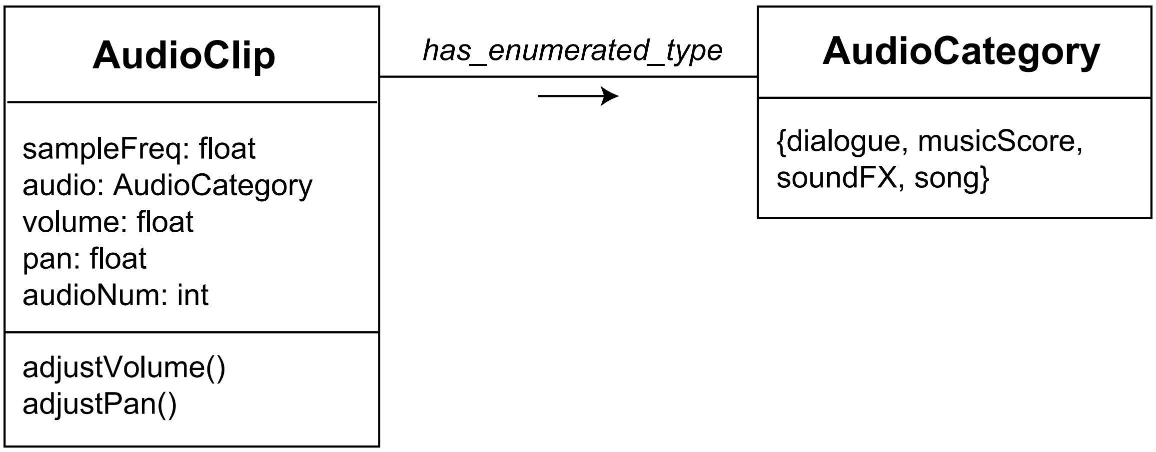


**Figure 15:** The **AudioClip** class with a link to **AudioCategory**

The **AudioClip** class has five unique attributes: (1) *sampleFreq*, which is the sample frequency of the audio in kiloHertz (e.g. 44.1); (2) *audio*, which is of the enumerated type **AudioCategory**, and which describes the category of audio for the *AudioClip* instance; (3) *volume*, which refers to the decibel level of the audio; (4) *pan*, which refers to the stereo distribution of the sound; and (5) *audioNum*, which, like *clipNum* and *textNum*, is used as an ordering number during playback. The two operations serve to manipulate the *volume* and *pan* attributes.

### 7.7.8 *The Entire OOIC Class Structure*

Below is a summary of all the classes and their relationships in the OOIC model as described in Sections 7.7.1 to 7.7.7.

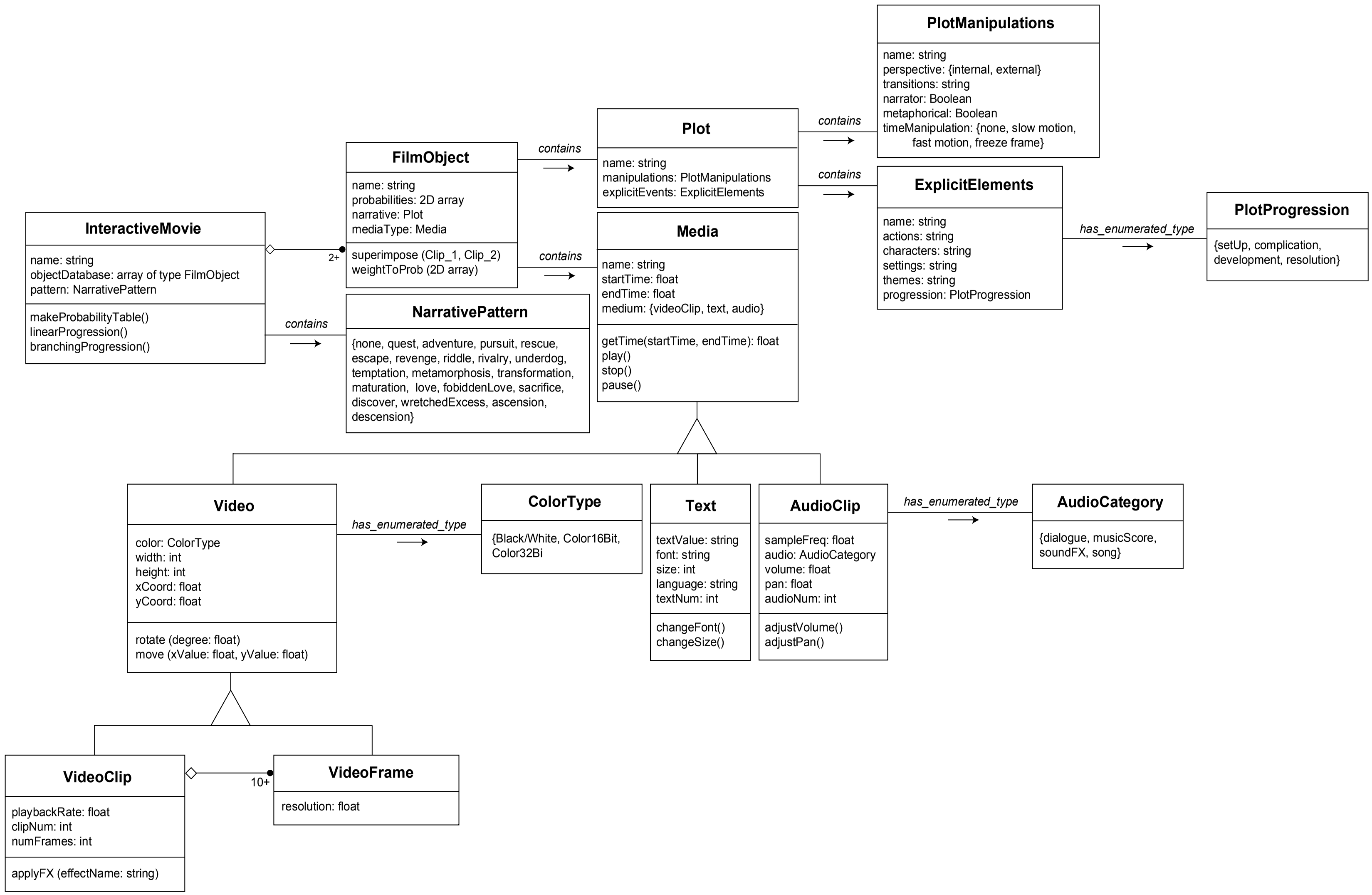


**Figure 16:** The OOIC class structure

### *7.8 Montage*

> *"As theorized by Vertov, film can overcome its indexical nature through montage, by presenting a viewer with objects that never existed in reality"* (Manovich 2001, p.149).

Montage is an important concept to Object-Oriented Interactive Cinema, even though the concept of montage was developed in traditional filmmaking. The difference between montage and compositing, according to how I am defining them, is that compositing refers to the juxtaposition of live-action material with computer-generated graphics, and montage is the juxtaposition of different shots, or video objects (of class **Video**), in time and/or space, regardless of how they were made.

Many film theoreticians believe that montage is the key twentieth-century technology for creating fake realities (Manovich 2001). There are two basic kinds of montage: temporal and spatial. Temporal montage is what most people think of when referring to montage, and involves the juxtaposition of separate realities to form consecutive moments in time. Spatial montage involves the merging of separate realities into a single image, such as split screen or superimposition. Spatial montage is also referred to as "montage within a shot" by Eisenstein (1969) in *Film Form: Essays in Film Theory*.

Russian writer Yury Tynyanov compared montage to prosody (the study of versification). He argued that temporal montage forces shots into a system of rhythmic equivalences which invite the reader to compare meaning across the film units in a similar manner to the effect the break has between lines in verse (Bordwell 1993). The Russian film director Lev Kuleshov also had strong opinions about montage:

> "For Kuleshov, montage was the essential factor differentiating cinema from the other arts and forming the basis of the specific impact that film can make. In a series of informal experiments, he showed that editing could create emotions and ideas not present in either of the single shots." (Bordwell 1993, p.121).

Russian filmmaker Sergei Eisenstein was instrumental in initiating concepts and theories around montage in cinema. In discussing rhythmic audiovisual montage, he states that each shot in a film has an accent, or that a shot is made up of stronger and weaker moments (Eisenstein 1969). Eisenstein believed that the role of the filmmaker was to balance the rhythmic accents of the visual track with the accents of the audio track (Bordwell 1993). For example, juxtaposing a strong visual accent against a weak audio accent will produce a certain effect, whereas pairing strong audio and visual accents together at the same time will produce a shot of high intensity. These theories show how Eisenstein was not only thinking about visual montage, but that he was also interested in exploring the relationship between sound and image through audiovisual montage.

### *7.8.1 Examples of Temporal Montage*

- The flashback
- The jump cut
- Parallel cutting
- Cross-cutting
- The eye-line match or a kind of precognitive flash forward cutting as used by Nicholas Roeg.

### *7.8.2 Temporal Montage in OOIC*

Temporal montage is controlled algorithmically in Object-Oriented Interactive Cinema. An effective way of organizing and managing film objects in the interactive movie database is to assign a number to each object's *clipNum* attribute using a sequential numbering system. These numbers, as well as any other attribute including the film object's name, can be used to control temporal playback.

Temporal montage is typically designed using conditional statements. For example, given two film objects named Clip_1 and Clip_2, and the condition that Clip_1 must be played before Clip_2, the following algorithm (written in *Python*) may be called if the program tries to play Clip_2:

```
if hasPlayed(Clip_1):
     play(Clip_2)
else:
     play(Clip_1)
```

In the above example, the hasPlayed() function returns a Boolean value (true or false) depending on whether Clip_1 has been played already or not. If not, the 'else' condition is followed and Clip_1 is played instead of Clip_2.

The above code can be modified to test the *clipNum* attribute instead of testing based on the film object's name. For example, say the Clip_1 film object actually has a *clipNum* value of 4, and the Clip_2 object has a *clipNum* value of 3, and the only thing we are interested in is that a film object with a lower *clipNum* value is played first, the following code may be written:

```
if Clip_1.clipNum > Clip_2.clipNum:
     play(Clip_2)
else:
     play(Clip_1)
```

In this example, dot notation is used to access the *clipNum* attribute of each of the film objects. If Clip_1's *clipNum* value is greater than Clip_2's (as it is in this case, since 4 is greater than 3), then Clip_2 gets played, otherwise Clip_1 gets played.

### *7.8.3 Examples of Spatial Montage*

- Superimposition:

Perhaps the first example of a film with superimposition is Vertov's *Man with a Movie Camera:*

> "And this is why Vertov's film has particular relevance to new media. It proves that it is possible to turn 'effects' into a meaningful artistic language" (Manovich 2001).

Superimposition in the context of OOIC involves the collage of two or more film objects together on the screen at the same time. A superimposition can be applied to a video, text, or audio object, although the implementation of the function would be different in each case. As seen in Section 7.7.2, the **FilmObject** class has a superimposition function:

```
superimpose(Clip_1, Clip_2)
```

The product of this superimposition could then be put through the function again to create a collage with three film objects, and so on, limited only by the capacity of the computer.

```
collage_1 = superimpose(Clip_1, Clip_2)
collage_2 = superimpose(collage_1, Clip_3)
collage_3 = superimpose(collage_2, Clip_4)
```

- Split-screen montage:

Split-screen montage is where the movie is divided into two or more movie windows, with no overlap or superimposition between the windows. Each window contains its own movie, although typically each movie relates to the other(s) in some way. The classic example is the split-screen technique used to show two people talking on the telephone.

Examples of linear movies which use split-screen montage include:

1. *Timecode* (2000).
2. *Woodstock* (1970).
3. *The Longest Yard* (1974).
4. *The Thomas Crown Affair* (1968).
5. *The Boston Strangler* (1968).
6. *Napoleon* (1927)
7. *More American Graffiti* (1979).
8. *Man with a Movie Camera* (1929).

Split-screen montage in OOIC simply involves the presentation of two or more non-overlapping film objects on the screen at the same time (limited, however, to objects of media type **VideoClip**). A function for having two video film objects presented one on top of the other could look like the following (again, written in *Python*):

```
def splitScreen(Clip_1, Clip_2):
     """
     This function puts two film objects on the screen at
     the same time, one on top of the other, given an 800
     x 600 screen resolution.
     """

     # Position and size of Clip_1
     Clip_1.xCoord = 0.0
     Clip_1.yCoord = 0.0
     Clip_1.width = 400
     Clip_1.height = 300

     # Position and size of Clip_2
     Clip_2.xCoord = 0.0
     Clip_2.yCoord = 300.0
     Clip_2.width = 400
     Clip_2.height = 300
```

The `def` simply means that a function is being defined, and the `"""` and `#` indicate comments which are not part of the executable code.

- Multiple Screen Montage:

Cinema has traditionally been presented as a single movie on a single screen. Multiple screen projections are nothing new, but digital technology allows for multiple screen interactive cinema, which is something currently only being explored by interactive artists and experimental filmmakers. Given the constraint of one screen, filmmakers have customarily been focused on experimenting with single screen montage, developing such techniques as ellipsis and condensation. "It's not that these things go away when deploying multiple screens but the distribution of images spatially complicates the intensity of such strategies and grammars as they are deployed in parallel." (Lafia 2003). The multiple screen interactive movie requires the filmmaker to re-evaluate the meaning of montage.

Multiple screen interactive movies are within the framework of OOIC. Under the OOIC model, the interactive nature of the multiple screen environment develops from the temporal and spatial manner in which the different film objects are displayed – the temporal and spatial manner in which the film objects are displayed are affected and dependent on the interaction of the participant with the system's input devices (e.g. sensors).

A multiple screen interactive movie involves two or more instances of the **InteractiveMovie** class. Each *InteractiveMovie* object is controlled by the same program, which directs each movie to the appropriate screen.

# Chapter 8:
# OOIC Design – Creating a Multi-Linear Narrative

## *8.1 Selection as Authorship*

The Object-Oriented Interactive Cinema approach to moviemaking is an example of selection as authorship. The implication of selection as authorship is that the author is no longer the creator of the original content, but that by allowing the viewer control over content and content order (temporal/spatial organization), a new author emerges: the spectator-author (Bathes 1977; de Certeau 1984). “The creative energy of the author goes into the selection and sequencing of elements rather than into original design” (Manovich 2001, p.130). As David Rokeby states about his experience with interactive technologies, “meaning was contingent upon my current thought patterns, my personal frames of references, [and] my own cultural baggage” (Rokeby, 2003).

Thus, in the OOIC model, the audience is the author because it is their interaction with the program controlling the film objects that creates the narrative. Their ‘selections,’ whether through conscious user inputs or through inputs unknown to the user, construct the cinematic text. This chapter illustrates how graph theory and transition matrices can be used as a construct for ‘selection as authorship’ in the OOIC model.

## *8.2 Object-Oriented Design*

### *8.2.1 Transition Networks*

Transition networks can be used to abstract the design process. A transition is a sequence of identifiable and classifiable changes or behaviors to an object. Objects undergoing transitions are said to be *active*, and its attribute values or behavior at a particular moment in time is called its *state*. “A transition is a binary directed connection between pairs of states" (de Champeau 1993). The OOIC design model describes the possible patterns of objects, attributes, and relationships (links) that can exist in a system.

An *event* is a particular impetus from one object to another that happens at a point in time. An event received by an object can induce a change of state in that object, or the sending out of another event to either the original sender or a different object.

> "The pattern of events, states, and state transitions for a given class can be abstracted and represented as a state diagram. A state diagram is a network of states and events, just as an object diagram is a network of classes and relationships." (Rumbaugh et al. 1991).

**Notation:**

The graphical notation in OOIC for a *state* is a circle.

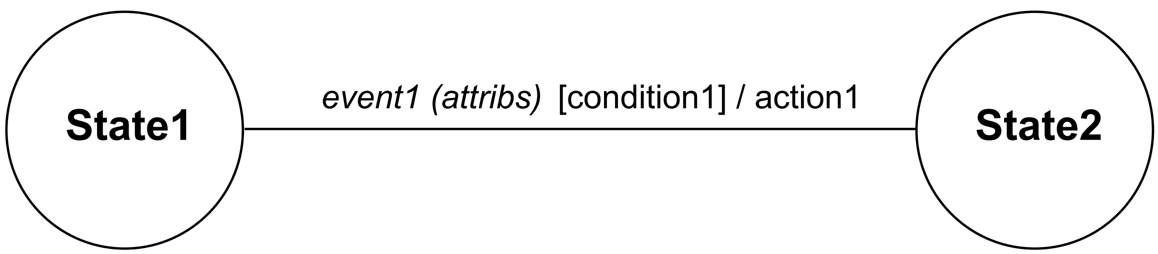


**Figure 17:** A transition state diagram

As can be seen from the message between the states in Figure 17, sometimes a transition has a condition attached to it. A condition of this type is called a *guard*. "A guard is a condition (Boolean-valued function) that must be satisfied in order for the transition to occur" (de Champeau 1993). An *action* is an operation associated with an event, and is executed virtually instantaneously compared to the execution of the state diagram. The notation for an operation on a transition is a slash ('/') followed by the name of the operation. Events can be grouped into event classes. The data values conveyed by an event are its attributes.

The idea of a cinematic element having a state is discussed by Murray (2000), although she does not make the connection to formal transition networks as described by object-oriented design and graph theory.

> "We can give an item in a frame a state... and make decisions based on that state without having to respecify everything we know about the general category and the particular object in order to make that decision. We can also store with our knowledge of a particular item the instructions for how to interact with it..." (Murray 2000).

As can be seen from the above quote, Murray's ideas fit quite logically within an object-oriented paradigm, with a frame analogous to an object, and a frame's state analogous to an object's attribute values and behavior at a particular moment in time. This comparison helps to emphasize my belief that applying an object-oriented methodology is the next logical step to research in interactive cinema.

### *8.2.2 The Dynamic Model*

An OOIC installation, upon first set up and before any user interaction, represents the *static* structure of the system. The static structure is the state of the system's objects and their relationships to each other at a single moment in time (Rumbaugh *et al*. 1991). As the program runs and the objects are manipulated by the user, the object states change, and these aspects of the system which are influenced by time make up the *dynamic model*. "The state of an object encompasses all of the (usually static) properties of the object plus the current (usually dynamic) values of each of these properties... The state of an object represents the cumulative results of its behavior" (Booch 1994).

Events represent external stimuli. A user's interaction with the OOIC system instigates events, which leads to state transitions. The mapping of these events and state transitions is the dynamic model. Using the dynamic model it is possible to map a scenario. "A scenario is a sequence of events that occurs during one particular execution of a system" (Rumbaugh *et al*. 1991). A scenario is analogous to an instance diagram.

A state diagram describes the behavior of a single class of objects. In OOIC design, a state diagram is typically modeled around the film objects (instances of the **FilmObject** class) that make up the interactive movie. “A state diagram is a graph whose nodes are states and whose directed arcs are transitions labeled by event names” (Rumbaugh *et al*. 1991). In order to fully understand the meaning of nodes and directed arcs, the next section begins a formal introduction to graph theory.

### *8.3 Graph Theory and the Multi-Linear Narrative*

Graphs, trees and matrices can be effective tools for constructing and visualizing multi-linear narratives. A possibility tree, for example, which “shows all possible outcomes of a multi-step operation with a finite number of outcomes for each step” (Epp 1995, p.602), can be drawn for any multi-linear narrative that uses a branching structure (i.e. embedded transformative interactivity). An embedded narrative structure is defined in Section 3.3.3 as a finite world where there are a quantifiable number of possibilities available to the end-user, which is quite analogous to the above definition for a possibility tree. Although drawing parallels between these two concepts may at first seem trivial, it is important to realize that I am attempting to apply formal mathematical terminology to the discussion of multi-linear narrative. It is my goal to demonstrate that once we understand the relationships that can be drawn between graph theory and multi-linear narrative constructs, we can start to use mathematical concepts and theorems to aid us in authoring multi-linear stories.

Why model an interactive cinema narrative using graphs? I don’t believe I could express it any better than the following quote by Marie-Laure Ryan:

> "For the semiotician, a plot is a type of semantic structure – and as a spatial configuration of elements, a structure is most efficiently represented in a visual model. For the specialist in artificial intelligence the answer is more practical: a graph is an object that can be handled by a computer. An adequate system of graphic representation is therefore a prerequisite to the simulation of the mental processing of the narrative text. The purpose of the plot-graph is to capture the reader's internalization of the narrative message of the text, the way plots are stored in memory." (Ryan 1991).

## *8.4 Graphs*

As math is based on definitions, throughout the rest of this chapter I outline a number of mathematical definitions that are relevant to the discussion.

***Definition 1:***

> "A **graph** G consists of two finite sets: a set V(G) of **vertices** and a set E(G) of **edges**, where each edge is associated with a set consisting of either one two vertices, called its **endpoints**" (Epp 1995, p.603).

***Definition 2:***

> "A **directed graph**, or **digraph**, consists of two finite sets: a set V(G) of vertices and a set D(G) of **directed** edges, where each edge is associated with an **ordered pair** of vertices called its endpoints" (Epp 1995, p.607).

***Definition 3:***

> "A **weighted** graph is a graph for which each edge has an associated real number weight" (Epp 1995, p.685).

These three definitions give a starting point to begin discussing multi-linear narrative in the context of graph theory.

### *8.4.1 The difference between weight and probability*

A **probability** value is a chance percentage, represented as a decimal number. For example, a probability value of 0.25 for a particular transition means that there is a 25%

chance that the transition will occur, all other factors being equal. The total probability values for a given film object must add up to 1.0. A **weight** is a number that is used in the *calculation* of probability values. A weight is changed into a probability value through the following equation:

***Equation 1:***

```
probability value = weight / sum of all weights
```

For example, given the film objects *Object_1*, *Object_2* and *Object_3*, and a weight of 2 for the transition from *Object_1* to *Object_2*, and a weight of 1 for the transition from *Object_1* to *Object_3*, the probability of going from *Object*_1 to *Object_2* is calculated as follows:

```
probability value (Object_1 to Object_2) = 2 / (2+1)
probability value (Object_1 to Object_2) = 0.67
```

And the probability of going from *Object_1* to *Object_3* is calculated as follows:

```
probability value (Object_1 to Object_3) = 1 / (2+1)
probability value (Object_1 to Object_3) = 0.33
```

Thus, there is a 67% chance of going from *Object_1* to *Object_2*, and a 33% chance of going from *Object_1* to *Object_3*. Notice that the probability values for *Object_1* all add up to 1.0.

Using weights instead of probabilities has the advantage that the total number of transitions and their associated probability values do not need to be known ahead of time, since when dealing with probability values the sum total must be 1.0. For example, a transition from *Object_1* to *Object_4* with a weight of 2 could be added during run-time, and the probability values would simply be recalculated to reflect the new 'sum of all weights' value. The probability of going from *Object_1* to *Object_2*, in this case, would change automatically to 0.4 (2 divided by 5).

### *8.4.2 The* probabilities *Attribute*

In a *FilmObject* instance the *probabilities* attribute is an array of actual probability values, not weights. This means that if weights are used, the **FilmObject** class function weightToProb() must be used to construct a proper *probabilities* array. Figure 18 recalls the **FilmObject** Class:

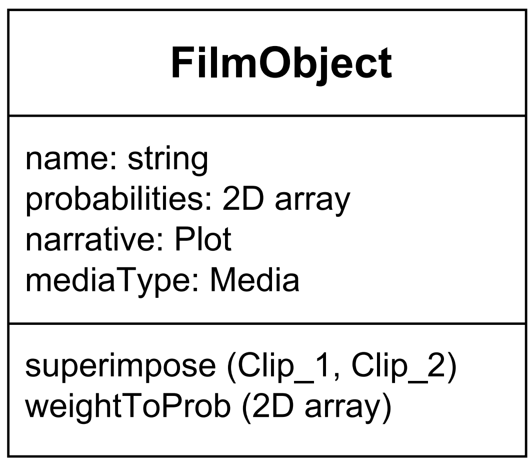


**Figure 18:** The **FilmObject** class

The weightToProb() function turns weights into a probability values using Equation 1. The weightToProb() function takes a 2-dimensional array as a required argument, and so the recommend procedure when using weights is to build a separate 2-dimensional array with weight values, and then apply the weightToProb() operation. The following code illustrates how this could be implemented for the *FilmObject* instance *Object_1*:

```
weightsArray = [['Object_2', '3'], ['Object_3', '2'],
['Object_4', '2']]
Object_1.probabilites = weightToProb(weightsArray)
```

The above code also illustrates the organization of the *probabilities* attribute. The format for each array element in a *probabilities* array is:

```
[name of destination object, probability value]
```

Likewise, the format for each array element in an array of weights should be:

```
[name of destination object, weight value]
```

Using the above code, the value of the *probabilities* array for Object_1 would be [['Object_2', '0.42'], ['Object_3', '0.29'], ['Object_4', '0.29']]. This can be translated into: The probability of going from *Object_1* to *Object_2* is 42%, and the probability of going from *Object_1* to either *Object_3* or *Object_4* is 29%.

In the discussion below I sometimes refer to weights instead of probabilities for film object transitions, but in the actual instantiation of a *FilmObject* instance, these weights are transformed to probabilities using the weightToProb() function.

### *8.4.3 Graphs Continued*

In my description of the "Re-Waking Life" installation in Chapter 10, I mention that if a user triggered an input that had no pre-programmed one-to-one relationship, an algorithm ran that chose the next film object to be played based on a probability table. For example, given the *FilmObject* instances Clip 4, Clip 5, Clip 20, Clip 30, and Clip 35, if Clip 4 was the currently playing film object, there was a weight of 2 for each transition to Clip 5, Clip 30, or Clip 35, and a weight of 1 that the next clip would be Clip 20. This sentence is hard to adequately visualize, however, and so by turning these relationships into a weighted graph we end up with a better model for knowledge representation.

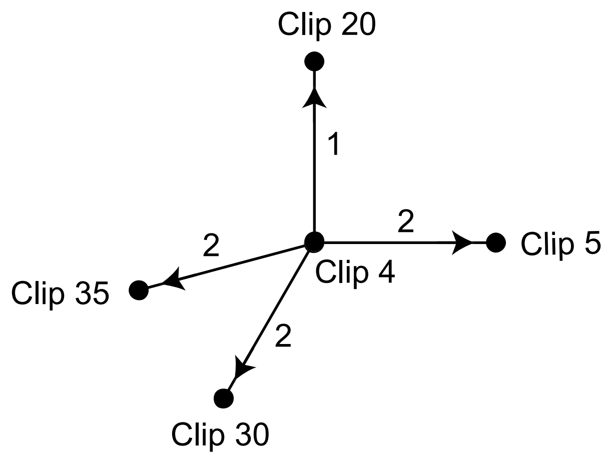


**Figure 19:** A weighted digraph

The instance diagram for Figure 19 would be as follows:

**(FilmObject)**
Clip 4
[['Clip 5', '0.286'], ['Clip 20', '0.142'], ['Clip 30', '0.286'], ['Clip 35', '0.286']]

**Figure 20:** An instance diagram for Figure 19

Going one step further by adding in the weights for one of the terminal vertices in Figure 19, we end up with the following **connected** digraph:

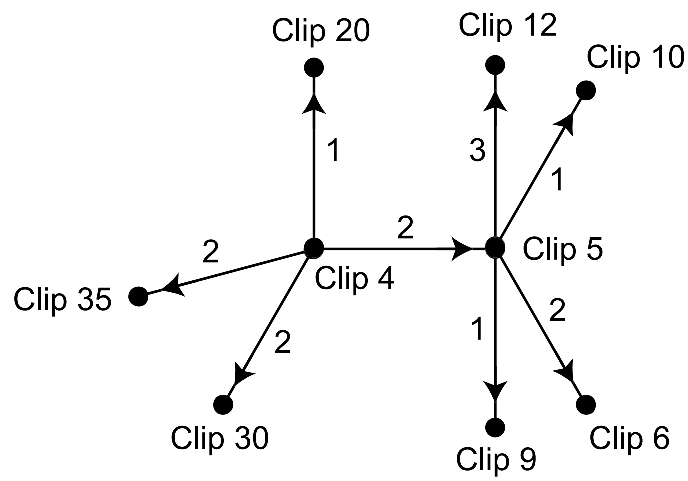


**Figure 21:** A connected weighted digraph

The equivalent instance diagram is represented by Figure 22:

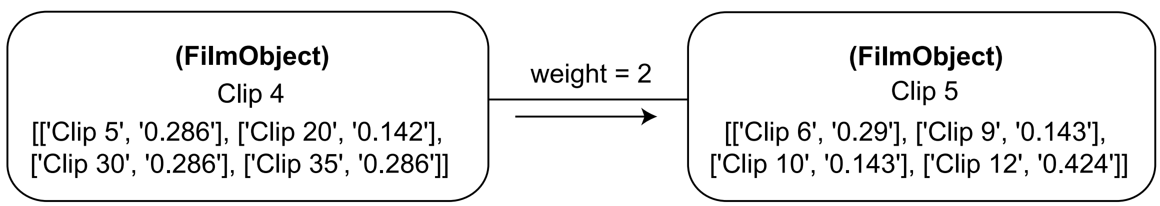


**Figure 22:** An instance diagram for Figure 21

Using Figure 21, more mathematical definitions can be introduced that can be applied to multi-linear narrative composition..

***Definition 4:***

"A **walk**… is a finite alternating sequence of adjacent vertices and edges" (Epp 1995, p.621).

***Definition 5:***

"A **path**… is a walk… that does not contain a repeated edge" (Epp 1995, p.622).

***Definition 6:***

> "A **closed walk** is a walk that starts and ends at the same vertex. A **circuit** is a closed walk that does not contain a repeated edge " (Epp 1995, p.622).

As can be seen from Figure 21, one possible **path** begins at Clip 4, goes to Clip 5, and then proceeds to Clip 10. This **walk** is a **path** through the film object database. In Figure 21 there are no closed walks and thus no circuits, so we also know that none of the objects have the possibility of being repeated.

Many interactive narratives, especially classical hypertext novels and *Choose-Your-Own-Adventure* books, take the form of a branching tree structure, and as trees are also defined by graph theory, further definitions can be introduced to the working model.

***Definition 7:***

> "A graph is said to be **circuit-free** if, and only if, it has no nontrivial circuits. A graph is called a **tree** if, and only if, it is circuit-free and connected." (Epp 1995, p.664).

If we assume that we are starting from Clip 4, Figure 21 can be redrawn as a tree:

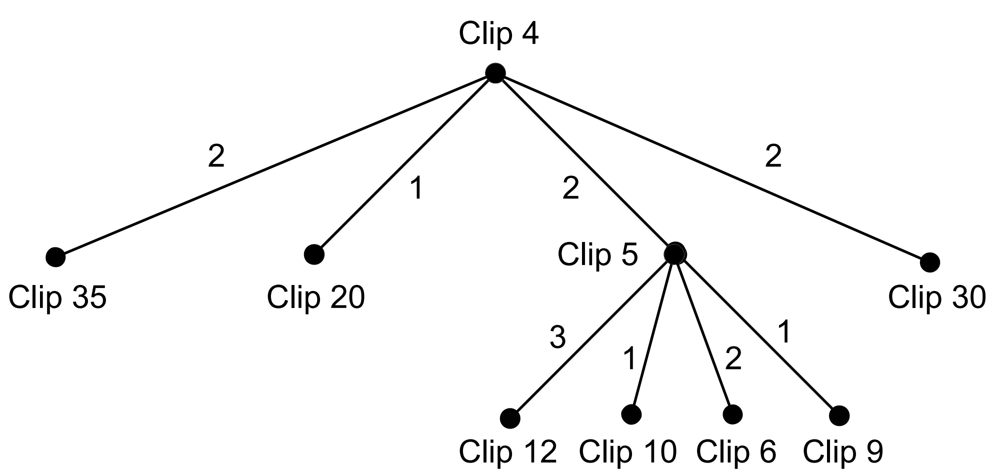


**Figure 23:** A weighted tree

In Figure 23, Clip 35, Clip 20 and Clip 30 are all **terminal** vertices at level 1; Clip 12, Clip 10, Clip 6 and Clip 9 are **terminal** vertices at level 2; Clip 5 is an **internal** vertex; and Clip 4 is the **root** (level 0).

The instance diagram for Figure 23 is as follows:

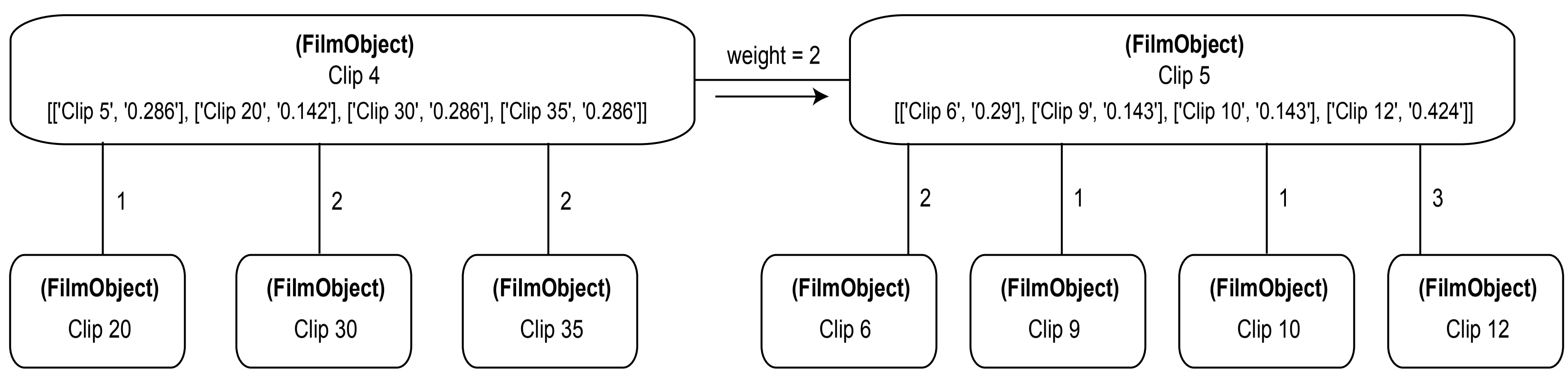


**Figure 24:** An instance diagram for Figure 23

If Figure 21 is redrawn again to include circuits, it is no longer considered a tree as per Definition 7:

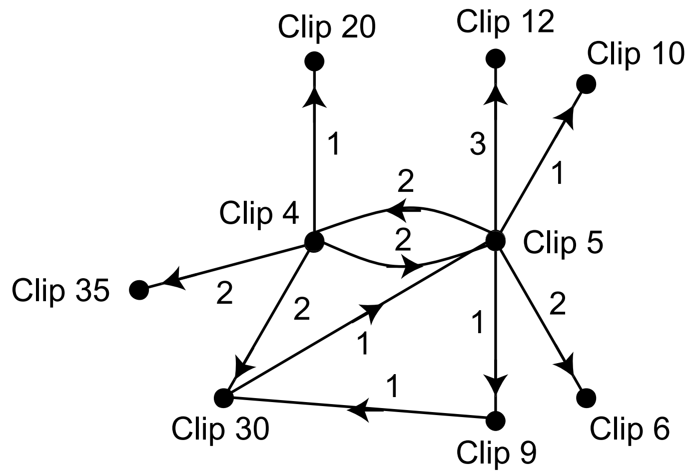


**Figure 25:** A weighted digraph with circuits

Figures 23 and 25 bring up some interesting concepts for multi-linear narrative construction. Most people are familiar with the types of narratives that can be made from a tree structure. In these situations, the final path that the user takes is ultimately a circuit-free walk through the narrative, which can also be interpreted as an essentially linear story progression. Stated another way, in a tree-structured narrative, although the narrative construction is multi-linear, giving the user agency over the story's direction, the user experiences the story as a linear progression (unless one takes into account repeat readings). This is not necessarily an undesirable thing in multi-linear storytelling, however. In fact, for my "Re-Waking Life" installation, I configured the weighted probability program to generate a tree structure based on which film clips had previously been played. For example, if Clip 4 was playing and the user triggered a changeClip() behavior, whereby the program accessed the probability table and changed the movie to Clip 5, the program would then alter the probabilities for every film clip in the database such that the probability of going back to Clip 4 was zero. In this way I managed to avoid constant repetition of film objects with higher associated probabilities. This algorithm effectively ensured that there were no circuits created within the changeClip() function, and since all objects were connected, the resultant narrative path met the definition for a tree.

A transition state diagram can be created for the above example:

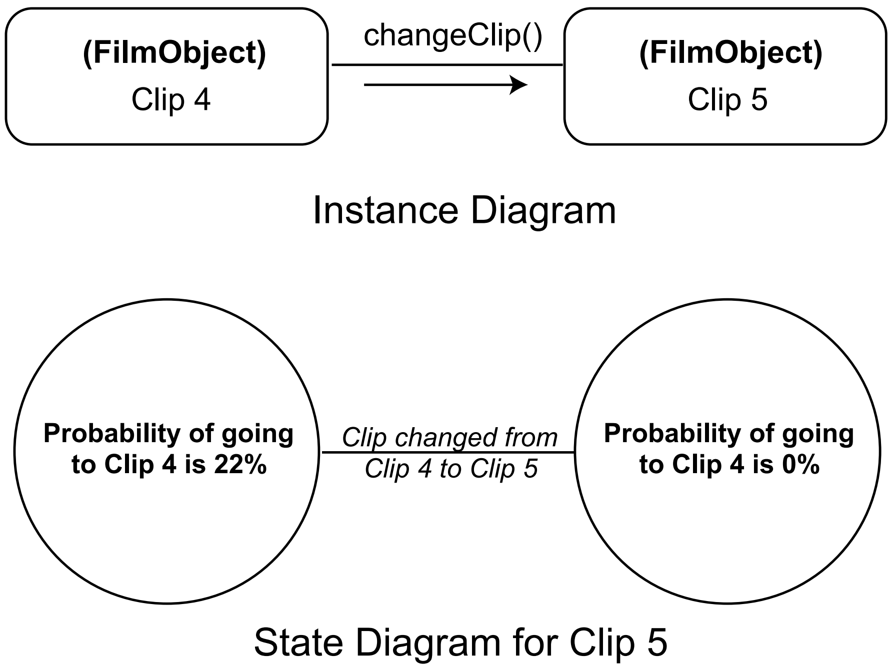


**Figure 26:** An instance and state diagram for the changing of Clip 4 to Clip 5

In actually, however, the changeClip() function was not the only algorithm running that could have changed the presently playing movie. There were pre-programmed user actions, for example, that had a direct one-to-one relationship to a particular film object or behavior. A user could thus cause a repeat screening of a particular film object, making a narrative circuit analogous to Figure 25. This model for multi-linear narrative construction is interesting because it forces us to think about storytelling in a more modular, or object-oriented fashion. Having film objects (seen as nodes or vertices in graph form) that can repeat themselves introduces a new type of story where the film objects themselves – and also in their repetition – contribute to the overall narrative. Under the OOIC model, it is not necessary for a film object to repeat itself, but the *possibility* or *ability* of a film object to repeat itself, or overlay on top of itself in a spatial montage, is fundamental.

### *8.5 Matrix Representation of Graphs*

So far I have only described very simple multi-linear narrative graphs, whereby I graphically represented one or two vertices (film objects) and their associated directional paths. In order for a narrative to be interesting, engaging, and merit extended and repeat viewings, however, a much more complex system is needed. For "Re-Waking Life," I had 36 film objects (vertices) and 164 weighted directional walks entered into a probability table. This number of vertices and directed edges would not be practical to represent as a graph, and as it happens, "all the information needed to specify a graph can be conveyed by a structure called a matrix" (Epp 1995, p.640). Matrices are the conceptual equivalent to two-dimensional arrays, and are the way that graphs are represented inside a computer. The *probabilities* attribute of the **FilmObject** class is shown in the class diagram to be a 2-dimensional array, but it is also valid to see it as a matrix.

***Definition 8:***

"An $m$ x $n$ (read 'm by n') matrix **A** over a set $S$ is a rectangular array of elements of $S$ arranged into $m$ rows and $n$ columns" (Epp 1995, p.640).

$$A = \begin{bmatrix} a_{11} & a_{12} & . & a_{1n} \\ a_{21} & a_{22} & . & a_{2n} \\ . & . & . & . \\ a_{m1} & a_{m2} & . & a_{mn} \end{bmatrix}$$

Given a directed graph, an *adjacency matrix* can be constructed. An adjacency matrix is the term to describe a matrix that represents a directed graph, where the values of the matrix are the number of arrows from $ai$ to $aj$, for all $i, j$ = 1, 2, … $n$. An adjacency matrix can be constructed for Figure 25 as follows:

*Let Clip 4 = a1, Clip 5 = a2, Clip 6 = a3, Clip 9 = a4, Clip 10 = a5, Clip 12 = a6, Clip 20 = a7, Clip 30 = a8, and Clip 35 = a9.*

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

**Figure 27:** An adjacency matrix of the directed graph in Figure 25

The best way to interpret the matrix is to read rows by columns. The rows, from top to bottom, and the columns, from left to right, represent *a1*, *a2*, ... *a9*. Thus, it can easily be seen from matrix **A** that there is a walk from Clip 5 (*a2*) to Clip 12 (*a6*), but there is no way of getting to Clip 20 (*a7*) from Clip 10 (*a5*). In fact, since the *a5* row is all zeros, there is no way of getting to any other film object from Clip 5. Clip 5 is thus a terminal vertex. The top-left to bottom-right diagonal of the matrix, called the *main diagonal*, is made up of all zeros, meaning that there are no trivial loops, or that a film object cannot be repeated immediately after itself. All this information can be very useful when constructing a multi-linear narrative. By representing all the possible narrative paths as a matrix, one can easily see where there might be holes in the narrative (e.g. if a column for a given clip in **A** was all weight 0, that clip could not be accessed from any other clip), or narrative dead-ends (e.g. a clip in **A** with a row of weight 0 would represent a terminal vertex and the narrative would not be able to progress further).

The 36 film objects (vertices) and 164 weighted directional walks programmed into "Re-Waking Life" can be represented by the matrix below. Rather than construct an adjacency matrix, which would only show which objects are connected to other objects, the values in the matrix below are the weights associated with the vertex pairs (*ai*, *aj*) for *i, j* = 1, 2, … 36. A matrix value (*i, j*) that is greater than zero tells us two things: there is a directed edge from *i* to *j*, and that the edge is weighted according to its matrix value. For convenience and clarity, any cell with a weight of zero I have left blank. The numbers 1 through 36 at the top, bottom, and left represent the *clipNum* attribute values of the film objects and are not part of the matrix itself.

*Let Clip 1 = a1, Clip 2 = a2, Clip 3 = a3, … Clip 36 = a36.*

**B =**

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** | **32** | **33** | **34** | **35** | **36** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | | 3 | | | 1 | | | 1 | | | | | | | | | | | | 1 | | | | | | | | | | | | 1 | | | | 2 |
| **2** | | | 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 1 | | 2 | 1 | |
| **3** | | | | 3 | | | | | | | | | | | | | | | | | | | | | | | | | 3 | | | | | | | |
| **4** | | | | | 2 | | | | | | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | 2 | |
| **5** | | | | | | 2 | | | 2 | 1 | | 2 | | | | | 1 | | | | | | | | | | | | | | | | | | | |
| **6** | | | | | | | 2 | | | 1 | 1 | | | | | | | | | | | 1 | | | | 1 | | | | | | | | | | |
| **7** | | | | | | | | 2 | | | | 2 | | | | | 2 | | | | | 1 | | | 1 | | | | | | | | | | | |
| **8** | 1 | | | | | | | | 2 | | | | | | | | | | | 1 | | | | | | | | | | | | | 1 | | | |
| **9** | | | | | 1 | | 2 | | | 2 | | | | | | | | 1 | 1 | | | | | | | | | 1 | | | | | | | | |
| **10** | | | | | | | | | | | 1 | | | | | | | | 1 | | | | 1 | 1 | | 1 | | | | | 1 | 1 | | | | |
| **11** | | | | | | | | | | | | 2 | 2 | | | | | 1 | 1 | | | | | | | | 1 | | | | | | | | | |
| **12** | | | | | 1 | | 2 | | 1 | | | | 1 | | | | | | | | 1 | | 1 | | | | | | | | | | | | 1 | |
| **13** | | | | | | | | | | | 1 | | | 2 | | | | | | | | | | | | | | 1 | | | | | | | | |
| **14** | | | | 1 | | | | | | | | | | | 2 | | | | | | | | | | 2 | 1 | 1 | | | | | | | 1 | | |
| **15** | | | | 1 | | | 1 | | | | | 1 | | | | 1 | | | | | 1 | | | | | | | | | | | | | | | |
| **16** | 1 | | | | | | | | | | | | | | | | 1 | | | | | | | | | | | | 1 | | | 1 | | | | 1 |
| **17** | | | 1 | | 1 | | 1 | | | | | | | | | | | 1 | | | | | 1 | | | | | | | | | | | | | |
| **18** | | | | | | | | | 1 | | | | | | | | 1 | | 1 | 1 | | | 1 | | | | | | | | | | | | | |
| **19** | | | | | | | | | 1 | | 1 | | | | | | | | | 1 | | | | | | | | | | | | | | | | |
| **20** | 1 | | | | | | | | | | | | 1 | | | | | | | | 2 | | | 1 | | | 1 | | 1 | | | | | | | 1 |
| **21** | | | | | | | | | | 1 | | | | | 1 | | | | | 1 | | 1 | | | | | | | | | | | | 1 | | |
| **22** | | | | | | | | | | | | | | | | | 1 | | | | | | 1 | 1 | 1 | | | | | | | | | | | |
| **23** | | | | | | | | | | | | | | | | | | 1 | | | | | | 1 | | 1 | 1 | | | | | | | | 1 | |
| **24** | | 1 | | 1 | | | | | | | | | | | | | | | | | 1 | | | | 1 | | | | | | 1 | | 1 | | | |
| **25** | | | | | | | 1 | | | 1 | | | | 1 | | 1 | | | | | | | | | | 1 | | | | | | 1 | | | | |
| **26** | | | 1 | | | | | | | 1 | | | | | | | | | | | | | | | | | 1 | 1 | 1 | | | | | | | |
| **27** | | | | | | | | | | 1 | | | | | 1 | | | | | | 1 | | | | | | | 1 | | | | | | | | |
| **28** | | | | | | | | 1 | | | 1 | | | | | | | | | | | | | | | | | | 1 | | | | | | | 1 |
| **29** | | | | | | | | 1 | | | | | | | | | | | 1 | 1 | | | | | | | | | | 1 | | | | | | |
| **30** | | | | 1 | | | | | | | | | | | | | | | | | | | 1 | | | | | | | | 1 | | | | | |
| **31** | | | | | | | | | | | | | | | | 1 | | | | | 1 | | | | | | 1 | | | | | 1 | | | | |
| **32** | 1 | | | | | | | | | | | | | | | 1 | | | | | | | | | | | | | | | | | 1 | | | |
| **33** | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 1 | | 1 |
| **34** | | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 2 | |
| **35** | | | | | | | | | | 1 | | 1 | | | | | | | | | 1 | | | | | | | | | | | | | | | 1 |
| **36** | 1 | | | | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** | **32** | **33** | **34** | **35** | **36** |

**Figure 28:** A matrix representation of the film object weights in "Re-Waking Life"

A number of interesting points can be gathered from Figure 28:

1. There are no empty rows or columns, indicating that there are no terminal vertices (narrative dead-ends) or disconnected vertices (narrative holes). This

also means that if the program was let run without any changes to the matrix, the narrative would continue indefinitely.

2. Because the weights in the main diagonal (top-left to bottom-right) are all zero, a film object can never be repeated immediately after itself.

3. For every row $n$ there is a value in the ($n$+1) column of that row. These values form the diagonal that is column-offset by 1 from the main diagonal. This diagonal indicates that for every object $c$, there is at least the possibility that the next film object could be Clip ($c$+1). I implemented this because the film footage that I was using was originally constructed as a linear feature film (*Waking Life*), and so I imagined that the most logically coherent narrative would be that which is created from the playback of the film in its original order. Because of the number of other weights associated with each film object, however, the chance of the film playing back in its original order for very long was quite small.

A useful exercise from here would be to make two histograms from matrix **B**. One histogram would be for the row values, with the film objects plotted against the number of possible objects that they could make a walk to. This would give the multi-linear author some idea of the 'bushiness' inherent in the narrative's branching structure. The second histogram would be for the column values, with the film objects plotted against the number of other objects that could make a walk to it. This would be important if there was a particular object that the author felt was important to the narrative, and which the author wished to be easily accessed coming from other film objects.

### *8.6 The Transition Matrix*

The values of matrix **B** are weights, but as previously mentioned, in the actual instantiation of an *InteractiveMovie* instance – which in this case is the "Re-Waking Life" installation – these weights get translated into probabilities using the weightToProb() function discussed in Section 8.4.2. Through the weightToProb() function, the program replaces each weighted value with a *transition probability* value, obtained by dividing the particular weight by the sum of all the film object weights. In doing this, a *transition* or *stochastic* matrix is generated.

***Definition 9:***

A **transition probability** is "a conditional probability for the system to go to a particular new state, given the current state of the system" (Carter 1996).

***Definition 10:***

A **transition** or **stochastic matrix** is a square (*n x n*) matrix of probability vectors.

### *8.6.1 The makeTransitionMatrix() operation*

Recall the **InteractiveMovie** class, which has the makeTransitionMatrix() operation:

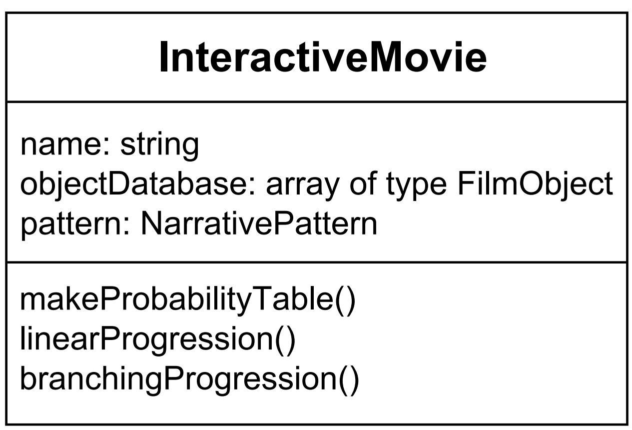


**Figure 30:** The **InteractiveMovie** class

The makeTransitionMatrix() operation assembles all of the *probabilities* attributes from each of the *FilmObject* instances contained in the *InteractiveMovie* object's *objectDatabase* array. The product of the function is a transition matrix for the *InteractiveMovie* instance. Each of the film objects in the transition matrix is a *state*, and an *absorbing state* is a film object from which we cannot get to any other film object (an absorbing state is equivalent to a terminal vertex).

Calling the makeTransitionMatrix() function on the "Re-Waking Life" *InteractiveMovie* instance produces a transition matrix. I have decided not to display the entire transition matrix in this thesis, but for the purpose of the discussion if have provided the first two rows:

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** | **32** | **33** | **34** | **35** | **36** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | | 0.33 | | | 0.11 | | | 0.11 | | | | | | | | | | | | 0.11 | | | | | | | | | | | | 0.11 | | | | 0.22 |
| **2** | | | 0.43 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0.14 | | 0.29 | 0.14 | |

**Figure 29:** The first two rows of the "Re-Waking Life" transition matrix

Each of the two rows in Figure 29 is a *probability vector*, and you will notice that they each add up to 1 (taking into account rounding off). It can be read from Figure 29 that if the changeClip() function was implemented in "Re-Waking Life" and the currently playing film object was Clip 2, there would be a 43% chance that the next object played would be Clip 3. A state diagram can be drawn to illustrate this:

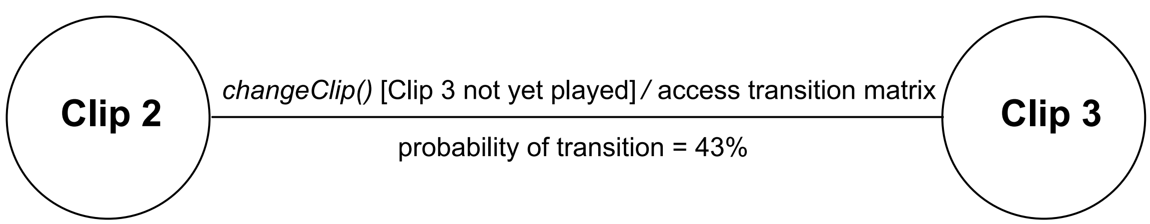


**Figure 31:** A state diagram for the "Re-Waking Life" *InteractiveMovie* object

From Figure 31 it can be seen that if the changeClip() function was called, a conditional guard would be evaluated (testing whether or not Clip 3 had been played already), and if not, the transition matrix would be accessed. From the transition matrix we learn that there is a 43% chance of going to Clip 3 given Clip 2.

### *8.6.2 Matrix Multiplication*

Let **C** = the transition matrix for the “Re-Waking Life” *InteractiveMovie* object, generated from the makeTransitionMatrix() function. Matrix **C** is referred to as a “1-step probability transition matrix,” where **C**(*i, j*) is the probability of going from state *i* to state *j* in one step. This type of matrix is also referred to as a first-order *Markov chain*[15], meaning that a transition is made based on the probabilities associated with the current state (i.e. the last decision made) (Cope 2000; Dodge and Jerse 1997).

Using transition matrices, a number of other mathematical calculations become available that can convey information about the narrative flow, or about how the narrative objects relate to one another. Using matrix multiplication, for example, we can determine the probabilities for transitions from one object to another in *n*-number of steps (describing an *n*th-order Markov chain).

***Definition 11:***

If **A** is a transition matrix, then for each positive integer *n*,

the *ij*th entry of $A^n$ = the probability of going from state *i* to state *j* in *n*-number of steps, for all integers *i*, *j* = 1, 2, … *m*.

Definition 9 tells us that if we found the resulting matrix from taking the square of matrix **C,** each value of that matrix would represent the probability of going from Clip *i* (*ci*) to Clip *j* (*cj*) in 2 steps, for all integers *i*, *j* = 1, 2, … 36. Calculating the cube of matrix **C** would give us a matrix representing the probabilities in 3 steps, and so on. These kinds of calculations would be useful, say, if it was important to the narrative that

1)

[15] Markov chains can be described according to their order. Where a zero-order Markov chain involves a completely random decision, an *n*th-order Markov chain makes a choice based on the conditions prescribed by the previous *n* decisions.

particular narrative objects were closely related to each other. A high probability value for going from state *ci* to state *cj*, for example, in a low number of steps, would indicate that Clip *j* has a good chance of being played – and within a relatively small amount of transitions – should Clip *i* be made the current object.

For an example, say an *InteractiveMovie* object was instantiated with a database of film objects, and the enumerated type for **PlotProgression** was the Aristotelian Arc categories *{setUp, complication, development, resolution}* as described in Section 7.7.4. The author might want to construct a narrative transition matrix such that the probability of going from an object with a 'setUp' *progression* attribute to an object with a 'complication' *progression* attribute in 2 steps is greater than 50%; the probability of going from an object with a 'complication' *progression* attribute to an object with a 'development' *progression* attribute in 3 steps is greater than 60%; *etc*. Using Definition 11 is one way that a multi-linear author could attempt to compose a multi-linear narrative using traditional story mechanisms (e.g. an Aristotelian Arc) in conjunction with graph theory and matrix multiplication.

Conditional transitions (Markov chains) can be effective tools for not only programmatically manipulating OOIC objects, but also for comparing the determinacy between different interactive movie instances. As Cope states in relation to his work with computer-generated music, "describing algorithmic programs according to Markov probabilities can be useful as a comparison tool, not just in terms of sophistication but in terms of complexity, hierarchy, and integration" (Cope 2000, p.38). For example, an interactive movie that is controlled using first-order Markov chains is produced relatively randomly compared to a more deterministically produced interactive movie using sixth-order Markov chains.

### *8.7 The Pseudo-Generative Narrative*

As most interactive multi-linear narratives would be quite complex, requiring a great deal of planning, programming, and composition, I believe that the most useful mathematical concept discussed in this chapter is the *transition* or *stochastic matrix*. Digraphs may be useful for visualizing one particular story node to see how it relates to the rest of its connected nodes (i.e. vertices), but this would be a limited and unique case. Tree diagrams could be useful, but mostly for those types of multi-linear narratives that had a fixed root state and which did not allow for circuits to be formed. Trees are also very effective visualization tools for branching-type interactivity. An adjacency matrix of the entire narrative network could also be useful, for one could easily see which nodes had walks to each other, and one could also look for narrative dead-ends and gaps, although the same amount of information – and more – could be obtained from a matrix with the weighted values. And a matrix with the weighted values is only one step away from a transition matrix, where the weights are turned into transition probability values.

A transition matrix not only tells us which story nodes are connected, but it also tells us the probability of going to a particular story node given a starting state. From this information we can easily see which nodes are more likely, or less likely to be accessed, and thus we can get a holistic understanding of how the multi-linear narrative could unfold. Having said this, there can still be advantages to building the weighted matrix, for the values of the transition matrix may be long decimal numbers, and it is easier to scan a large matrix for patterns using integer values (which is typically how weights are expressed).

The main advantage of using a transition matrix is that it allows us to introduce some randomness to the narrative flow, thereby giving the interactivity a pseudo-generative feel. I call it *pseudo-generative* because we are still dealing with a finite number of possibilities, and a true *generative* narrative would allow for a countless or unquantifiable number of possible story paths.

### *8.7.1 The linearProgression() Operation*

One of the great advantages of using a transition matrix for multi-linear narrative composition and analysis is its flexibility. Even if an author wished to compose a strictly linear narrative, this can still be easily accomplished in the OOIC model – it simply implies a particular construction for the transition matrix. A transition matrix which describes a linear narrative progression means that the diagonal which is +1 offset from the main diagonal is all of value 1.0, and the rest of the matrix probability values are zero. For example, given an InteractiveMovie instance with 10 film objects, the following transition matrix **A** describes a linear narrative assuming that the *a1* film object is the first object to the played:

$$A = \begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

**Figure 32:** A matrix describing a linear narrative progression

The linearProgression() operation of the **InteractiveMovie** class takes a transition matrix made from the makeTransitionMatrix() function and changes the probability vectors to the format shown in matrix **A**, where the diagonal which is +1 offset from the main diagonal is all 1.0, and all other values are zero. This causes the film objects to be played back in a linear order from the first film object in the *objectDatabase* array to the last.

### *8.7.2 The branchingProgession() Operation*

If the multi-linear author did not wish for the narrative path to be determined in any way through stochastic processes, but a forking narrative path was still desired, a transition matrix is still relevant, for it would just imply that all of the values in each probability vector would be equivalent. For example, given the connected states 1, 2, 3, and 4, and beginning at state 1, there would be an equal chance of going to state 2, 3, or 4, with the path's outcome determined only by the user's direct input. This is the effect of the branchingProgression() operation of the **InteractiveMovie** class. Essentially this function takes the transition matrix made from the makeTransitionMatrix() function and distributes all of the probability values equally for each probability vector. For example, given an *InteractiveMovie* object with three film objects, assume that the following transition matrix **B** was created from the makeTransitionMatrix() operation:

$$B = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.35 & 0.45 & 0.20 \\ 0.10 & 0.60 & 0.30 \end{bmatrix}$$

Applying the branchingProgression() function to this matrix would produce the following matrix **C**:

$$C = \begin{bmatrix} 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \end{bmatrix}$$

### *8.7.3 The Stochastic Process*

By allowing for some randomness, however, and providing non-equivalent probability values to each of the states, the narrative flow becomes no longer finitely predictable based on one-to-one relationships between the user and the system. Although stochastic processes have the potential to disrupt or challenge the narrative coherency, they would make repeat viewings more enjoyable, for the narrative would rarely unfold the same way twice. Narrative coherency is, in fact, the biggest obstacle for multi-linear storytelling, whether it uses stochastic processes or not, and this is why I suggest in Section 3.2 that perhaps the only way around this hurdle is to re-think the narrative.

The reinvention of narrative, or finding new ways of constructing enjoyable and compelling narratives, is one of the principles of the Object-Oriented Interactive Cinema model. It is my belief that graph theory, and in particular transition matrices, are valuable tools for the multi-linear narrative author, especially one working in the OOIC model. Other artists have used similar mathematical constructs, such as *n*th-order Markov chains and augmented transition networks (ATNs) – most notably David Cope in his Experiments in Musical Intelligence (Cope 1996) – but OOIC is the first attempt that I am aware of to propose applying stochastic processes to cinema.

"Re-Waking Life," which was built using the OOIC model, was an experiment to demonstrate that a database of film objects, controlled through a transition matrix, can produce an engaging and unique narrative and cinematic experience. "Re-Waking Life" is described in more detail in Chapter 10.

# Chapter 9:
# Building a Language for an Interactive Cinematic Environment

Part of the design section of the OOIC model includes specifications for building interactivity into a cinematic environment. This chapter discusses these specifications by building a 'language' for interactive cinematic environments.

> "A technology is interactive to the degree that it reflects the consequences of our actions or decisions back to us. It follows that an interactive technology is a medium through which we communicate with ourselves... a mirror." (Rokeby 2002).

## *9.1 Computer Intelligence*

> *"Computer intelligence is defined as the ability of a computer to understand, reason, and apply knowledge. This involves sensing the environment, assimilating what is happening within that environment, and responding."* (Lovell 2002c).

In order to create interactivity, a system for knowledge representation and computer intelligence are needed. Knowledge is represented under the OOIC model through the class structure and through the algorithms that makes decisions based on sensory input. The class structure is the computer's *a priori* knowledge about the system; the information-capture and decision-making algorithms are the system's contextual knowledge. Computer intelligence is expressed through the effectiveness of the algorithms that capture and interpret the sensory data, and which manipulate the system's objects.

According to Lovell, there are three kinds of abilities that are required for computer intelligence in the interactive arts: *Perception*, *Reasoning*, and *Dexterity* (Lovell, 2002d). Perception is how well the computer understands the environment which it is trying to analyze through its sensory mechanisms, which can include video images, infrared sensors, proximity sensors, spoken words, text, UI interactions (such as mouse/keyboard input), *etc*. Reasoning is the computer's ability to interpret the sensed information in a meaningful way, and to make decisions based on the acquired data. Finally, dexterity is the computer's ability to provide a feedback response to the user/performer based on the interaction with the system. Responses involve the electronic manipulation of media, such as changes in lighting, sounds, visuals, robotics, *etc*. In the OOIC model, a response is a manipulation of one or more objects, typically of the **FilmObject** class.

> "The issue of who is controlling whom becomes blurred. The intelligence of the human interactors spreads around the whole loop, often coming back in ways they don't recognize, causing them to attribute intelligence to the system." (Rokeby 1998).

In complex, well-designed interactive systems, the goal is that the computer has enough intelligence to facilitate emergent or pseudo-emergent phenomena. Emergent and pseudo-emergent phenomena provide interactive artists with dynamic behaviors that are generated from the program's algorithms themselves, and produce results that go beyond the system's predictable responses.

### *9.2 A Media Structure*

The steps that can be used to build an interactive media space are outlined by Robb Lovell in his paper "A Blueprint for Using a Reactive Performance Space" (Lovell 2002a). Lovell encapsulates these categories in the term *media structure*. Below is a slightly modified version of Lovell's media structure; it describes the information feedback loop that occurs between the interactor and the system in an interactive space:

- **Action:** The physical phenomena, such as fingers typing on a keyboard or the movement of a person's limbs, which are used by the system as user input.
- **Sensing:** The digitization of the action, captured from a peripheral device such as a keyboard or a camera sensor.

- **Processing:** The transformation of the digitized data into meaningful units of information. Processing involves the algorithmic interpretation of the input data.
- **Translation:** The transformation of the interpreted information into a response decision based on the system's control program.
- **Generation:** The system's response to the user, based on the decisions made in the translation stage.
- **Presentation:** The part of the system that physically produces the media that the user experiences, such as a computer monitor or projector.

The next step in this process is the **reaction** of the user to what is presented, which can be treated as an **action**, and thus the feedback loop commences once more.

A project built on the OOIC framework would deal with each of the above media structure categories. So far in this thesis the discussion has been around analysis and design for OOIC, which involves the *Processing*, *Translation*, and *Generation* categories of the media structure. This chapter focuses on the first two media structure categories: *Action* and *Sensing*, as well as having some overlap with *Processing*. The final category, *Presentation*, as it relates to OOIC, was discussed in Section 4.5: The Screen.

### *9.3 Capture and Analysis*

There are generally four categories of media that can be controlled by a computer in an installation space: visuals, light, sound, and mechanical systems. Technologies to sense and interpret events that occur within the installation space include "video-based sensing, tracking systems, sound samplers, pitch detection, and analog sensors (heat, touch, bend, acceleration, *etc*)" (Lovell 2002b).

*9.3.1 Video-based sensing*

> "The environment will, as a given, be ambiguous in nature. The information contained within a video image is incomplete and limited, and since this is the computer's view of the world, its representation of the current state of its surroundings will be unreliable." (Lovell 2002c).

The task of having the computer understand what is happening in the physical world is a difficult one, for items contained in a scene are just blobs to the computer. Fortunately, these blobs have a location, size, velocity, and other measurable characteristics – characteristics that can be used for interactivity. But the task is never trivial. Special effects filmmakers get around some of the obstacles of video tracking by matching points on a custom-designed wearable suit to a 3-D model in the computer, but this is not practical for a general audience. While computers today cannot approach the level of perception of the human body, they can understand the outside world in a limited way through image processing techniques, through knowledge representation, and through assumptions about the physical environment (Lovell 2002c).

A major conceptual hurdle for artists and programmers dealing with video-based sensing is that a camera does not see the world the way humans do. Distance information is very difficult to measure, and as such, "actions that cut across the camera's view appear different than actions that move towards or away from the camera" (Lovell 2002b). Lighting is a crucial factor when doing video-based sensing. You need to be very aware, and have as much control as possible, over the lighting conditions of the installation space. How light falls on the people and props in the space determines how they will be seen by the computer.

Because of the great challenges with live video capture, many interactive installations use programs where "responses are based upon randomized manipulations or heuristic road maps for the computer to follow" (Lovell 2002c). The definitive challenge, then, is to be able to take the interactivity that one step further, to create a reactive space based on highly meaningful user interactions.

### *9.3.2 Information Extraction*

Because of the challenges in getting a computer to interpret a scene in the desired manner, the scene must be well controlled. This means that no strict rules for interpretation can effectively be established, but rather that each environment is unique, and that video-based sensing is heavily determined by context.

> "The person creating the means for a computer to understand part of an environment must make assumptions about the structure and content of the environment in order to create algorithms to extract information for the computer to use" (Lovell 2002b).

**Extraction Techniques:**

1. Motion: "Motion is calculated by subtracting successive images from each other, and counting the number of pixels that have changed... Under constant lighting conditions, motion is the change in surface area of objects in the scene" (Lovell 2002b). An object that is closer to the camera will appear to have more motion than the same object farther away from the camera because being closer to the camera causes many pixels to be affected.
2. Presence: To detect presence is to detect the simple presence or absence of light.
3. Background: A common practice with video-based sensing techniques is to grab a snapshot of the background without any objects in it to use as a baseline for comparisons. This is effective at showing foreground objects more clearly provided the foreground objects are not the same color and intensity of the background.
4. Objects: This technique tries to distinguish single entities within the camera's view. In order to do this, the objects need to appear different in some way to the computer. The most common methods are by having high contrast objects (e.g. light colored objects against a dark background), or by color tracking. Once an object is identified, it can be measured for traits such as motion, speed, location, *etc*.

In an article on human body tracking, Wren *et al*. (1998) describe how they used a combination of color and object detection to build a model for tracking people in an installation space:

> "The person model is built by first detecting a large change in the scene, and then building up a multi-blob model of the user over time. The model building process is driven by the distribution of color on the person's body, with blobs added to account for each differently-colored region. Typically separate blobs are required for the person's hands, head, feet, shirt and pants." (Wren et al. 1998).

Robb Lovell, a computer scientist and interactive artist, has created a software package called "EYES" (www.squishedeyeball.com), which is designed for video-based capture and analysis. The program works with MAX, a graphical programming language. Using "EYES," one is able to quite effectively track a number of distinct objects in a scene, and extract from them information such as location, speed, color, *etc*.

## *9.4 User Interaction*

> *"The interpersonal, back-channel communications and ancillary activities of the audience, which currently remain largely unsensed and unprocessed, can be just as important as the primary authored experience"* (Davenport 1998, p.19).

### *9.4.1 Human-Computer Interaction*

Human-computer interaction, or HCI, is a discipline concerned with the analysis, design, and implementation of how people interact with computers. HCI is often seen as a remediation of other media, both past and present. The Macintosh's famous GUI, for example, first introduced in 1984, uses the metaphor of an office desktop for the personal computer user interface – a metaphor that has since been adopted by Windows and become ubiquitous for current Windows and Mac operating systems.

The study of Human-Computer Interaction is crucial to the development of a successful interface and navigation structure for interactive environments. A major obstacle to effective design in today's rapidly changing and expanding digital world is that too often software and hardware engineers fail to adequately address how their audience will interact with their product. One of the most important issues in bridging the gap between man and machine is developing a user interface (UI) that is intuitive and easy to learn. Since the OOIC model is based upon human participation with a computer system towards the construction of an interactive multi-linear narrative, HCI and UI issues are important considerations.

### *9.4.2 Principles of User Design*

The following discussion outlines five principles for user design. The first principle is that the design for an interactive cinema project should be user-centered. As Mitchell Kapor states in his "Software Design Manifesto":

> "If a user interface is designed 'after the fact' it is like designing an automobile dashboard after the engine, chassis, and all other components and functions are specified" (Kapor 2002).

The important questions in creating a user-centered design for OOIC are: Who are the users? What are the main functions that the user will need? Why does a user want to interact with the interactive movie? Is the movie's interactivity accessible and understandable by users of different experience levels? What is the most intuitive way that the user could interact with the computer program controlling the movie?

One key issue that designers and programmers alike often forget (or neglect) is that interface design incorporates many different disciplines: hardware and software engineering, ergonomics, psychology, sociology, linguistics, computer science, *etc*. As such, the second important design consideration is integration of knowledge and experience from all of the HCI-related disciplines.

The third design consideration is that the system should be thoroughly tested before release to ensure that it contains no bugs that will inhibit its function. The quickest way to inhibit user enjoyment is to create frustration over simple interface and navigation issues.

The fourth design factor is an issue of commodity: the interface and navigation of the interactive movie should be well suited to complement the functionality of the system. They should also complement the movie's narrative.

The fifth and final design consideration is less quantifiable, yet important nonetheless. The idea is that the design should be pleasurable to use, incorporating visual, aural, and functional aesthetics. Theoretically, Object-Oriented Interactive Cinema can incorporate interactivity that relates to any of the sensory modalities.

In summary, the five principles of user design that should be incorporated into creating an interactive cinema experience based on the OOIC model are:

1. The design should be user-centered.
2. It should effectively integrate the HCI-related disciplines.
3. The interactive system should be free of major bugs.
4. The interface and navigation should be well suited to the functionality of the system, as well as the narrative structure and content.
5. The system for interaction should be enjoyable to use.

### *9.4.3 Principles of User Navigation*

The following discussion outlines five principles for user navigation. One of the most important concepts in navigation is that it should be easily learned. If it takes too long to grasp the navigational flow of a program, the user will become frustrated and lose interest. An easily learned navigation structure is the first step in creating user satisfaction. Instructions can be useful, but the navigation concepts should be intuitive enough that an excessive amount of preparation is not necessary.

> "The interactive artist must strike a balance between the interactor's sense of control, which enforces identification, and the richness of the responsive system's behaviour, which keeps the system from becoming closed" (Rokeby 2002).

Another important concept to help the user maintain a sense of spatial orientation is to remain consistent. From the audience perspective, this means not only a consistent look and feel to the physical interfaces, but also that the interactive cinema presentation responds to similar senor data in similar ways. If a user waves their arms, for example, and this is a method of interaction with the piece, the movie should respond in a logical manner, and respond in the same manner if the user repeats the action (unless, of course, a sense of abstraction, confusion, or mystery is desired in the particular piece).

> "The interactor waves his hand to trigger a sound. He then waves again, in a similar manner, to find out if the same sound will be triggered again. If something else is heard, the interactor may conclude that the system does not function well, or that it isn't really interactive at all." (Huhtamo 2003).

A feedback mechanism is crucial for effective navigation: the user needs to feel that their actions have meaning. For an audience navigating an interactive cinema presentation, feedback could come as sound cues, visual cues, tactile cues, *etc*. A sound cue could be a noise generated when a user steps into a certain area of the environment. A visual cue could be that when a user steps into a certain area the movie changes in response to the user's position.

> "[T]he interactor can never have an absolute control over the system. Rather, he enters into an on-going and evolving dialogue, a 'cybernetic feed-back loop' without a final resolution." (Huhtamo 2003).

The navigational tools should be based on the goals of the user, meaning that they should appear in context and support the flow of the composition. For interactive storytelling it is important to have continuous interaction such that the system does not need to pole the participant for input, and where the interactivity is inherent to the story. In order for the user's experience to be highly immersive, "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" (Galyean 1995).

Does the navigational structure support users coming from different technical or cultural backgrounds? Along with meeting the user's goals, the navigation structure should also be appropriate and support the interactive environment. It is often desirable in an interactive installation to have a transparent interface because it allows the user to participate without having to consider their direct relationship with the underlying system, but no interface can be completely transparent. The most one can hope for is that the interface be so well integrated as to be *subconsciously accepted* by the interactor as transparent.

> "When an interface is accepted as transparent, the user and his or her world are changed; the transforming characteristics of the interface, no longer contained by a visible apparatus, are incorporated into the user and the user's world view" (Rokeby 2002).

To summarize, the navigation framework that should be used in designing an OOIC movie is based on the idea that the navigation should:

1. Be easily learned.
2. Remain consistent.
3. Provide feedback.
4. Appear in context.
5. Support the user's goals.

### *9.5 The User Experience*

What kind of experience does one wish a user to have in an interactive cinema presentation? What kind of experience does the user themselves wish to have? Cinema is a participatory medium, whether through fan culture, movie star icons, interpretation of the movie text itself, or the community of the movie theatre. "The cinema is born immediately as a social machine... One could just as well propose that it is the spectators who invent cinema" (Comolli in Tafler 1999).

> "Rather than creating finished works, the interactive artist creates relationships... Rather than broadcasting content, interactive media have the power to broadcast modes of perception and action." (Rokeby 2002).

Blom and Chaplin discuss ideas around the experiential body of knowledge (Blom and Chaplin 1988); I discuss these ideas below in the context of Object-Oriented Interactive Cinema. The first concept is *kinesthetic awareness*, meaning that in interactive cinema participants have the opportunity to explore meaning behind their own movements. For example, audience members may ask themselves some of the following questions: "What will happen to the movie if I twist my body around this way?" "Is there a different effect produced if I move just my head and try to keep the rest of my body still?" "What kind of movie content produces what kind of effects on my kinesthetic awareness?"

> "An interactive system can be seen as giving the user the power to affect the course of the system, or as interfering in the interactor's subjective process of exploration" (Rokeby 1998).

The second concept in Blom and Chaplin's theory is *phrasing*: "All movement contains innate rhythms and phrases which provide the magic ingredients in any of the performing arts" (Blom and Chaplin 1988, p.17). The third concept is *form*. Questions the participants could ask themselves include: "What kind of responses will I get from the movie if I focus on grouping my body movements into circular patterns?" The fourth concept is *relating* – this concept is explored by the participant discovering their relationship to the sensors, the movie projection and the projection surface. The fifth and final concept is *abstraction* – because the movements are not in any way choreographed, each time a participant enters the interactive cinema space they create a new narrative, a new unfolding, and a new cinematic experience.

Stephen Levinson (2001) talks about how "our thinking is fundamentally spatial." This concept relates to interactive cinema because there is an inherent spatial element to the user's relationship to the screen when participating in the interactive environment, especially when the interface is something more than a mouse and keyboard. In my "Re-

Waking Life" installation discussed in Chapter 10, for example, the user's placement and movement of colored objects in the physical space are mapped into the computer to produce interactivity with the database of film objects.

> "It is just because, for us, spatial knowledge is a matter of higher-level thinking... that we can be deeply intrigued and tantalized by the artful manipulations of space by which the architect and sculptor play on our minds." (Levinson 2001, p.70).

Another issue in designing a user-experience for interactive cinema is what role the audience plays in the story or plot. The story could be designed around a first-person perspective[16], where all of the film objects are shot from the audience's point of view, and the audience is a protagonist in the narrative. In the Object-Oriented Interactive Cinema model, however, I believe that taking a third-person perspective would produce the best results. This would allow for more control and flexibility in the cinematic composition, and ultimately produce a better film. To increase the immersive qualities of the experience, it is still beneficial to give the audience a role within the narrative, but one that only indirectly affects the plot. The idea of staying away from a first-person perspective may seem, at first, to contradict the goal for increased immersion, but there are other interactive cinema theorists who make a similar argument:

1)

[16] First-person or internal interactivity is also discussed in Section 3.3.4.

> “Most designers of cyber worlds remain committed to creating first-person experiences, which immerse the participant in an unknown world of authored action and consequence, despite the limited success of this form... [T]he first-person viewpoint loses meaning as soon as the participant steps back to a more distant experience.” (Davenport 1998, p.16).

### *9.6 Interactive Environment Guidelines*

> *“Some exciting work has taken place in sensor technology for music applications and in vision algorithms that can, for instance, read sign language. However, no one has unearthed a more general, universal language for gesture, and none may be forthcoming.”* (Davenport 1996).

The OOIC model is built around interactivity for a novice user domain (i.e. not performers). Therefore, there are certain precautions which must be acknowledged:

- You cannot predict what people are going to do. Therefore, you must build into the system a certain amount of acceptance that some people won’t be doing the ‘right’ things. One way to accomplish this is to limit the flexibility of the interactions.
- If you capture a great deal of subtle variations it’s hard for the audience to know how they are affecting the movie; if you have the system respond to larger and fewer variations, the audience understands better. More than 3 or 4 different things happening at the same time is too many for people to understand what affect they are having.

  > “It is difficult to sense interaction in situations where one is simultaneously affecting all of the parameters... The constraints provide a frame of reference, a context, within with interaction can be perceived.” (Rokeby 2002).

- The most reliable tracking techniques are based on detecting location, motion, velocity, and direction of travel. All of these except location are cyclical, meaning that they can reverse direction rapidly. For cyclical traits you usually have to average the values to get meaningful results.
- You need a comfort level for movements and gestures.

- There is typically a great deal of ambiguity to deal with; context-specific rules need to be made for each installation space. The more rules and constraints that are applied to the installation space, however, the narrower the sensory data flow into the computer.

  > "By increasing the amount of filtering that is applied in the perceptual process that the interactive system employs, the designer increases the reliability of the resulting information and therefore the unambiguity of control, but at the same time, the richness of that information is reduced" (Rokeby 2002).

# Chapter 10:
# OOIC Projects

This chapter describes the interactive installation project “Re-Waking Life,” which was designed using the Object-Oriented Interactive Cinema model.

## *10.1 Re-Waking Life*

*“Our truest life is when we are in our dreams awake.”*

– Henry David Thoreau

The purpose of the “Re-Waking Life” installation was to explore ideas and practices around Object-Oriented Interactive Cinema (OOIC) and multi-linear narrative. It was my goal to produce an interactive object-oriented digital cinema installation that asked questions about how the project’s format and representation affected an audience’s cinematic experience.

### *10.1.1 Technical Considerations*

“Re-Waking Life” used footage from the animated feature movie *Waking Life* (directed by Richard Linklater, 2001), but added interactivity and applied the Object-Oriented Interactive Cinema model to the film. Since the OOIC model uses a database of film objects to construct interactive content, to begin, I extracted the entire *Waking Life* movie from DVD and broke it up into 36 segments, or 36 *FilmObject* instances. These 36 film objects were loaded into a database by the computer when the installation was started.

**(InteractiveMovie)**
"Re-Waking Life"
Database of 36 *FilmObject* instances

**Figure 33:** An instance diagram for the "Re-Waking Life" interactive movie

The installation's main physical prop was a large foldout bed. The original *Waking Life* movie is about dreams, lucid dreaming, dream states, and life philosophy, and thus the installation space was decorated in an attempt to complement these themes. For example, there were stuffed-animal toys and a jester hat positioned on the bed pillows, and a toy cowboy and fireman hat were hung on the wall.

**Figure 34:** The "Re-Waking Life" installation set-up

At the bottom end of the bed was a purple blanket with fourteen felt squares distributed across its surface – twelve blue squares and two green ones. On each of the felt squares a word was written: *Dream*, *Destiny*, *Life*, *Philosophy*, *Experience*, *Change*, *Chaos*, *Society*, *Freedom*, *Time*, *Human*, *Lucid*, and two special keywords, *Play* and *Mix*.


**Figure 35:** The “Re-Waking Life” dream map

This blanket, with its felt squares and words, was the main interface for interacting with the movie that was projected on a screen directly in front of the bed. I called this interface the ‘dream map’.


**Figure 36:** A felt square on the 'dream map'

The fourteen words made up the **PlotProgression** enumerated type for the *progression* attribute of the **ExplicitElements** class.

| **PlotProgression** |
|---|
| {Dream, Destiny, Life, Philosophy, Experience, Change, Chaos, Society, Freedom, Time, Human, Lucid, Play and Mix} |

**Figure 37:** The **PlotProgression** enumerated type used in "Re-Waking Life"

Positioned above the 'dream map' in the installation space was a wide-angle low-light camera. The image from the camera was used by the computer to create interactivity.

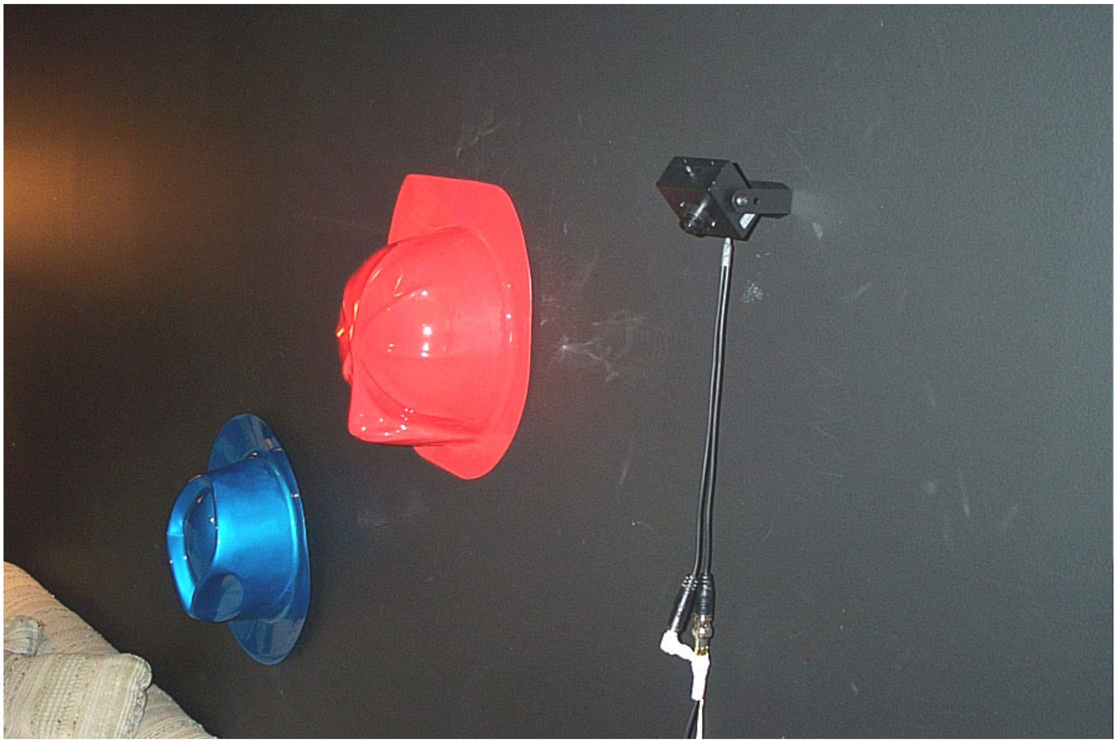

**Figure 38:** The low-light camera used to capture user interactions

The way the interactivity worked was that the interactor used two stuffed-animals – a yellow *Tweety Bird* and a white polar bear – to create relationship pairs using the words on the 'dream map'. The relationship pairs were interpreted by the computer to change the movie's content. For example, if one of the toys was placed on 'Lucid' and the other on 'Dream,' the computer responded by changing the currently playing movie clip (a film object) to one where the narrative content was about lucid dreaming. Word relationships were used to help infuse meaning into the interactivity.

Below is an instance diagram for one of the *FilmObject* instances to demonstrate the nested object attributes of the **FilmObject** class, and to show the typical structure for a film object in "Re-Waking Life." All of the film objects used in the installation had a *VideoClip* object for their *mediaType* attribute. Even though the film objects all had a *VideoClip* instance for their media component, they also had an audio element but the audio was part of the video track and thus not a separate **Media** object.

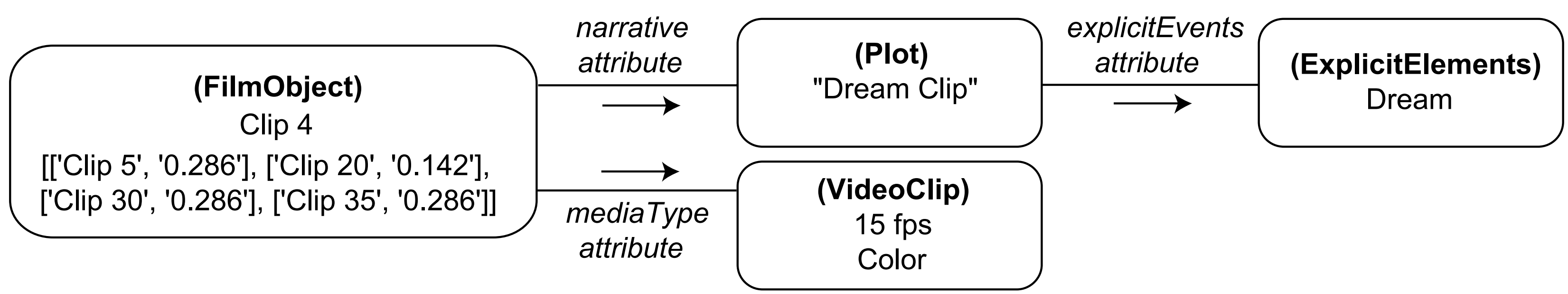


**Figure 39:** An instance diagram for the “Clip 4” *FilmObject* instance

Figure 40 below provides an alternate graphical interpretation of a film object instance to demonstrate the nested object attributes of the **FilmObject** class. Although this form of notation was not discussed formally as part of the OOIC model, I provide it here to show another way of drawing instance diagrams.

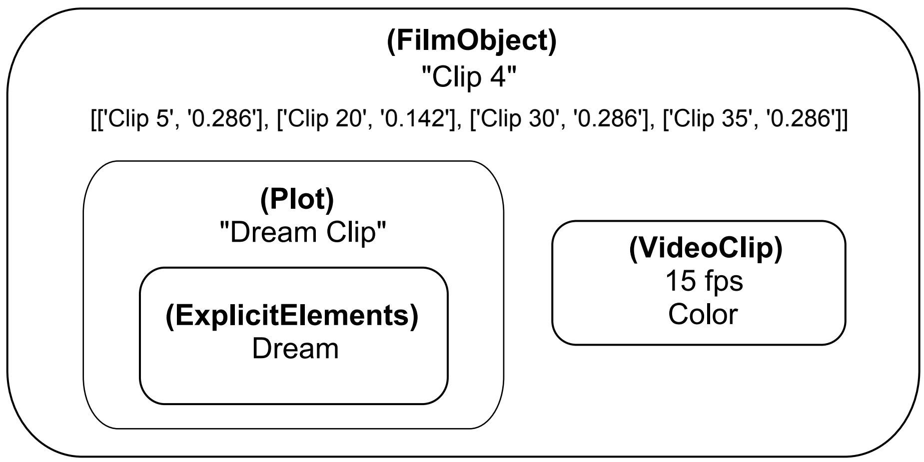


**Figure 40:** An alternate notation for the “Clip 4” *FilmObject* instance

### *10.1.2 Programming*

There were 47 word-pairs programmed into the system. Because there were a great deal more possible pairs possible using the 14 words, I implemented an algorithm to take effect should no pre-programmed pair be triggered. If the user placed the two toy animals on words which did not have a predefined relationship, the computer responded by accessing a table of probabilities and changed the clip according to a probability calculation. For example, if Clip 3 was the currently playing film object and a word-pair was made by the user with no established relationship, the computer looked up Clip 3 in the probability table and transferred to a new clip based on the probabilities associated with Clip 3. Clip 3 may have had a 30% chance of going to Clip 4, a 20% chance of going to Clip 10, a 20% chance of going to Clip 21, and a 10% chance of going to clip 35. To avoid repeating those clips with higher probabilities, once a clip was played, the probability of any clip transferring to it became zero, only to be reset once all of the 36 clips had been played or the algorithm reached a terminal transition probability (all possible transfers for the currently playing clip had been used up). The hope was that using a probability table in this way introduced a degree of randomness to the interactivity, and produced a pseudo-generative narrative. This aspect of the “Re-Waking Life” installation is also discussed in Section 8.5.

The two green felt squares had the words ‘Play’ and ‘Mix’ written on them, and indicated special functions. Instead of causing the movie to switch to a particular clip, relationship pairs that were formed with either of these two words caused the movie’s visual content to be affected. The ‘Play’ keyword was used to apply special effects to the currently playing movie using the applyFX() function of the **VideoClip** class (discussed in Section 7.7.6). For example, placing one toy on ‘Play’ and one on ‘Philosophy’ caused the movie to turn black and white. ‘Play’ and ‘Destiny’ caused an emboss effect to be generated, and ‘Play’ and ‘Time’ caused the movie to play at twice normal speed. The ‘Mix’ keyword was associated with collage – causing the superimpose() function of the **FilmObject** class to be implemented – and also with split-screen effects. Some relationship pairs caused two film objects to be played side-by-side on the screen at the same time (split-screen), whereas some word-pairs caused two film objects to be played layered on top of each other in a two clip collage. Some word pairs involving ‘Mix’ also affected the audio. For example, with one toy on ‘Mix’ and one toy on ‘Dream,’ a split-screen effect was produced with the audio heard only from the left-hand film clip; with ‘Mix’ and ‘Chaos,’ however, the audio was heard from both clips at the same time.

### *10.1.3 Capturing the Environment*

The interactivity in “Re-Waking Life” was controlled using Robb Lovell’s “EYES” software (www.squishedeyeball.com) and the MAX graphical (or dataflow) programming language (www.cycling74.com). Two Macintosh computers were used:

one to process the camera data, and one to control the playback and projection of the film objects based on the interpreted data.

The image that the camera captured of the blanket was divided into 14 sensor regions, each one positioned over one of the word-squares. The sensor thresholds were configured around color values – those of the two stuffed-animals. Thus if the polar bear was moved into a sensor region, the threshold would be crossed for detection of a white-colored object, and the computer would respond. The actual algorithm was much more complex than this, of course. For example, in order to filter out the movement of a stuffed-animal across one of the sensors when a user was moving it from one square to another, the computer only acted on a crossed sensor threshold if the movement within the sensor region was also less than a velocity threshold (indicating that the user had put the object down) for a given period of time (to eliminate the zero velocity of an object reversing directions). Also, to prevent overly rapid transitions, there was a slight delay between how quickly a new relationship pair could be activated once one relationship pair was successfully chosen.

### *10.1.4 Multi-Linear Narrative*

By using a database of film objects to interactively create content, each time a user sat down to experience “Re-Waking Life,” they constructed a different and unique narrative through the order and manner in which the film objects were juxtaposed together spatially and temporally (spatial and temporal montage). The method and flexibility through which the interactivity was implemented enabled pseudo-generative narrative construction, such that there were literally thousands of different narrative permutations or paths that a user could have followed through the film object database.

### *10.1.5 The User Experience*

The user's experience of narrative flow was affected by the order in which the film objects were presented, The 'feel' of the narrative was affected by how the user decided to implement and manipulate the visual effects (using the 'Play' square), and how the user decided to combine more than one clip together on the screen at the same time (using the 'Mix' square). Some interesting effects and narrative results could be produced through the spatial juxtaposition of two film objects, either in a side-by-side spatial montage, or in a directly overlapping collage. How is the narrative changed, for example, when a clip of a professor discussing the philosophy of the problem of free will is played concurrently with a clip of an angry man in a jail cell speaking profanities against those responsible for his incarceration? Is the effect different when the juxtaposition is displayed as two clips side-by-side versus as a collage? I believe that the "Re-Waking Life" installation raised many interesting and challenging questions around multi-linear narrative and the interactive cinema experience. I also believe that it was a successful implementation of the OOIC model, and proves how Object-Oriented Interactive Cinema is an effective framework for composing interactive cinema installations.

# Chapter 11:
# Conclusion

Storytelling is a fundamental part of our world. As children we love to have stories told to us by parents and friends, and as we get older we might turn to television and cinema to provide us with narrative stimulation. Stories capture our imagination and whisk us away on adventures into alternate realities; they make us feel happy and sad, frightened and at peace; stories help provide meaning to our existence.

This thesis has discussed a new form of storytelling, combining the pleasures of new media with the aesthetics of cinema's images and sounds. Object-Oriented Interactive Cinema is a model for constructing new ways of telling stories, allowing for new adventures into our imagination. OOIC gives us the pleasure of agency through interactivity, the richness of visual stimulation through its cinematic footage, and the excitement of music and dialogue from its sound compositions. OOIC provides a model for interactive cinema composers to create new forms of expression through the manipulation of new media objects.

## *11.1 The OOIC Composition Process*

Below is a summary of the OOIC architecture and composition process.

### *11.1.1 The Interactive Artist Skill Set*

Before embarking on the task of creating a provocative interactive cinema piece, the interactive artist needs to have certain skills and knowledge about the discipline within which they are situated. The interactive artist is someone who knows how to think procedurally, is computationally inclined, and knows how to use this knowledge to express their artistic intentions. They are someone who has a strong foundation in artistic practice and information technology. They are an artist-programmer, bridging worlds that have traditionally been on opposite ends of the academic spectrum. They are both left- and right- brain proficient.

### *11.1.2 Background Knowledge*

To effectively use the OOIC model, the author needs to have a good understanding of the principles of new media[17]. The principles can be summarized as follows:

1. Numerical representation.
2. Reproducibility.
3. Modularity.
4. Automation.
5. Dynamically generated content.

The author also needs to understand the principles of *interactive digital media*, which are the four principles that Janet Murray (2000) attributes to digital environments. I have chosen not to use the term *digital environments* in my discussion to emphasize the point that *interactive digital media* is a subcategory of new media, and that it describes properties that refer to particular new media *objects*. Interactive digital media characterizes artifacts which relate to a finer level of granularity (within the sphere of new media) than is implied by the word 'environment.' The four principles of interactive digital media are:

1. Interactive digital media is procedural.
2. Interactive digital media is participatory.

1)———————————

[17] See Figure 41 in the Appendix for a schema on how the domains of new media and interactive digital media, as well as the OOIC model, fit in with their related media spheres.

3. Interactive digital media is spatial.
4. Interactive digital media is encyclopedic.

Lastly, the author needs to understand the four different levels of interactivity:

1. Interpretive interactivity.
2. Utilitarian interactivity.
3. Designed choice interactivity.
4. Macro-interactivity.

The *interactive* in Object-Oriented Interactive Cinema (OOIC) refers to designed choice interactivity. It can be defined as a relationship between a user and a cinematic media object where the user has agency over the object's form and/or content, and thus actively contributes to the construction of meaning and/or influences the media object's response(s).

### *11.1.3 The Multi-Linear Interactive Narrative*

The next step in the composition process is for the author to establish a clear understanding of *narrative*, and what it means to make a narrative multi-linear and interactive. For these purposes, narrative can be divided into *plot* and *story* to distinguish those elements which are explicitly presented to us (*plot*), and the set of all events that the reader creates in their own mind from the presented plot elements (*story*) (Bordwell and Thompson 1997). In the OOIC model, the movie objects that are presented to the user (based on the user's interaction with the system) would make up the plot, and the resultant chronicle and experience that developed out of the user's interpretation of these objects, the story[18]. The process can be summarized as follows:

1)

[18] See Figure 42 in the Appendix for a diagrammatic flowchart of the sequence of events leading from author to story under the OOIC model.

1. The author, who has a particular story that they wish to present, builds an interactive system based on the OOIC model.
2. The user interacts with the system, and based on their social and cultural influences, the user makes choices. The system in turn, records and acts upon these choices. This step refers to Zimmerman's (2001) *designed choice* interactivity discussed in Section 2.6.
3. Based on the user's interactions with the system, a plot is created (consisting of that which is presented to the user).
4. The user witnesses the plot, constructs a story in their mind based on the experience, and reacts. The story that the user constructs influences the decisions they make in step 2. This step also relates to Zimmerman's (2001) *interpretative* interactivity discussed in Section 2.6.

The OOIC framework divides interactivity into six categories. The author should become familiar with each category and understand how each interactivity type is implemented[19]. The categories are:

1. External/exploratory
2. Internal/exploratory
3. External/embedded transformative
4. External/emergent transformative
5. Internal/embedded transformative
6. Internal/emergent transformative

Under the OOIC model, all of the interactivity types are feasible, although the degree of emergent transformative interactivity is limited. The OOIC model enables a small amount of narrative emergence through the use of stochastic processes and real-

1)

[19] See Figures 43 and 44 in the Appendix for two representational models of the OOIC interactivity framework.

time effects, but the underlying media content is embedded (being made up of pre-recorded material).

### *11.1.4 The Class Structure*

The first step in the actual composition process is to model the object-oriented class structure that will be used to construct the interactive movie. This is the object-oriented analysis (OOA) phase. In this thesis I have provided the framework for a class structure, but the author would certainly want to modify the architecture to suit the specific needs of their interactive film. Some classes, however, are fundamental to the OOIC model, and they should only be modified, not completely removed or changed. These classes are the **InteractiveMovie**, **FilmObject**, **Plot**, and **Media** classes.

The class structure is the scaffolding for the interactive movie composition. Attributes and operations can always be added as the authoring process progresses and more specific requirements are realized, but substantial changes to the base classes should be avoided. If the need for major restructuring becomes apparent, it is suggested that the author starts the entire class structure over again, perhaps by first redefining and clarifying the fundamental goals of the interactive movie. The finished class structure should be relatively straightforward, and provide a holistic view of the movie's architecture form both the perspective of its media and its discourse.

### *11.1.5 The Design Phase*

The first decision that must be made in the design phase is whether or not the movie's multi-linearity will involve stochastic processes or not. Will the narrative take on a pre-defined branching and/or looping structure, or will there be some randomness inherent to the narrative construction. There are pros and cons to both types of narrative flow, but they can both be modeled using graphs and transition matrices. A transition matrix is made by assigning each *FilmObject* instance a 2-dimensional array of probability values for transitions to other film objects, and then calling the makeTransitionMatrix() function of the **InteractiveMovie** class. A transition matrix describes the relationships between each of the film objects in the movie's object database. Algorithms can be used to manipulate the transition matrix to create different kinds of narrative flow. A transition matrix where all the probability values in a probability vector (matrix row) are equal implies a branching-type narrative. A matrix with different probability values means that some film objects have a greater chance of being accessed than others, and there is a weight to which way the narrative can flow. In this case, some randomness is introduced and the narrative path is not discretely predictable, making a pseudo-generative composition.

The **FilmObject** class has two included functions which can be applied to a transition matrix. A completely linear narrative can be made by calling the linearProgression() function, or a matrix with dissimilar probability values can be entered into the branchingProgression() operation to make each probability value equal for a every probability vector.

### *11.1.6 The Movie Environment*

Once the interactive movie's class structure is solidified, and design considerations are worked out, including the probability arrays for each of the *FilmObject* instances, the next stage is to figure out how the user will interact with the system's hardware and software. Will standard input/output devices be used, such as a mouse and keyboard, or is a more transparent user interface desired? How will the user navigate through the multi-linear content of the interactive movie, and are the navigation protocols intuitive and easy to learn?

This stage may require that the class structure be modified once it is determined what kinds of human-computer interfaces are desired. This is acceptable and, in fact, desirable since the composition process should consist of feedback loops between analysis, design, and implementation. If the classes themselves are accurate and well modeled, modifying particular attributes and behaviors should be easy to accomplish without disrupting the rest of the system.

#### *11.1.7 The Movie Content*

The final stage in the OOIC composition process is perhaps the most time consuming, but also allows for the most creativity. It involves actually writing a script for the interactive movie, storyboarding it, and then filming it. This is the final stage, but also the first, as the class structure should definitely be built with the movie's story content in mind. Essentially, one should have the basic story idea established before entering the object-oriented analysis and design phases of the OOIC model – the making of the movie's content (recording of sound and images) and the making of the movie's program (programming) should be intertwined and co-dependent processes, as each contributes to, and is affected by, the other.

### *11.2 The Future of Interactive Cinema*

There may come a day when computers have the capacity to hold and manipulate such a vast quantity of information that digital storytelling can become completely generative, or express fully emergent transformative interactivity. The narrative could be generated by matching user interactions with an enormous database of knowledge objects, each one containing a description of a story element – a setting, a character, an item, an event, a behavior, *etc*. The storytelling software could be programmed with such a high level of intelligence that it would be able to make an interesting, compelling, and unique narrative to suit each user's tastes.

There may come a day when computers have the ability to render and manipulate highly detailed 3-dimensional graphics in real-time, opening the floodgates to the promise of virtual reality as an entertaining and immersive medium. No longer would VR users be subjected to bulky head-mounted-displays, or have to strain to maintain their suspension of disbelief due to the poor graphics quality of the imagery. Every way the user looked the screen would change to match their point of view, and they just might feel like they are truly living and interacting in an alternate world.

There may come a day when the above-mentioned phenomena, the future of digital storytelling and the realization of compelling virtual reality, meld into a new media form. It would be VR with a narrative backbone. The *Holodeck* of *Star Trek* would be a reality. While this future form of *interactive digital media* is exciting to contemplate, it is not the type of interactive cinema that the OOIC model is designed to help create. While a virtual reality experience could be considered to be a form of interactive cinema – provided it used cinematic techniques in the manipulation of time and space – the OOIC model is focused on creating an interactive cinematic experience using pre-recorded material.

OOIC is modeled around the assumption that there are pre-existing cinematic media objects to be manipulated. According to the OOIC class structure, a cinematic object is either an audio object (i.e. dialogue, music, sound effects, *etc*.), an on-screen text object, or a video object (where images are presented through real or simulated camera angles). Thus, OOIC is a model that relies heavily on authored material, but where the material is then re-interpreted, re-sequenced, and re-appropriated by the user into an individual experience. OOIC is a compositional tool for an art form that tries to effectively combine cinematic objects with a system for interactivity, allowing for collaboration between the author and the user in the generation of a novel cinematic experience.

The Object-Oriented Interactive Cinema model provides a formal methodology for advancing interactive cinema research, demonstrating how object-oriented concepts can be used to effectively blend interactivity, multi-linear narrative, and film. OOIC is an artist-programmer's guide for creating a type of interactive cinema, and helps to bring the art form one step closer to its potential.

# Appendix

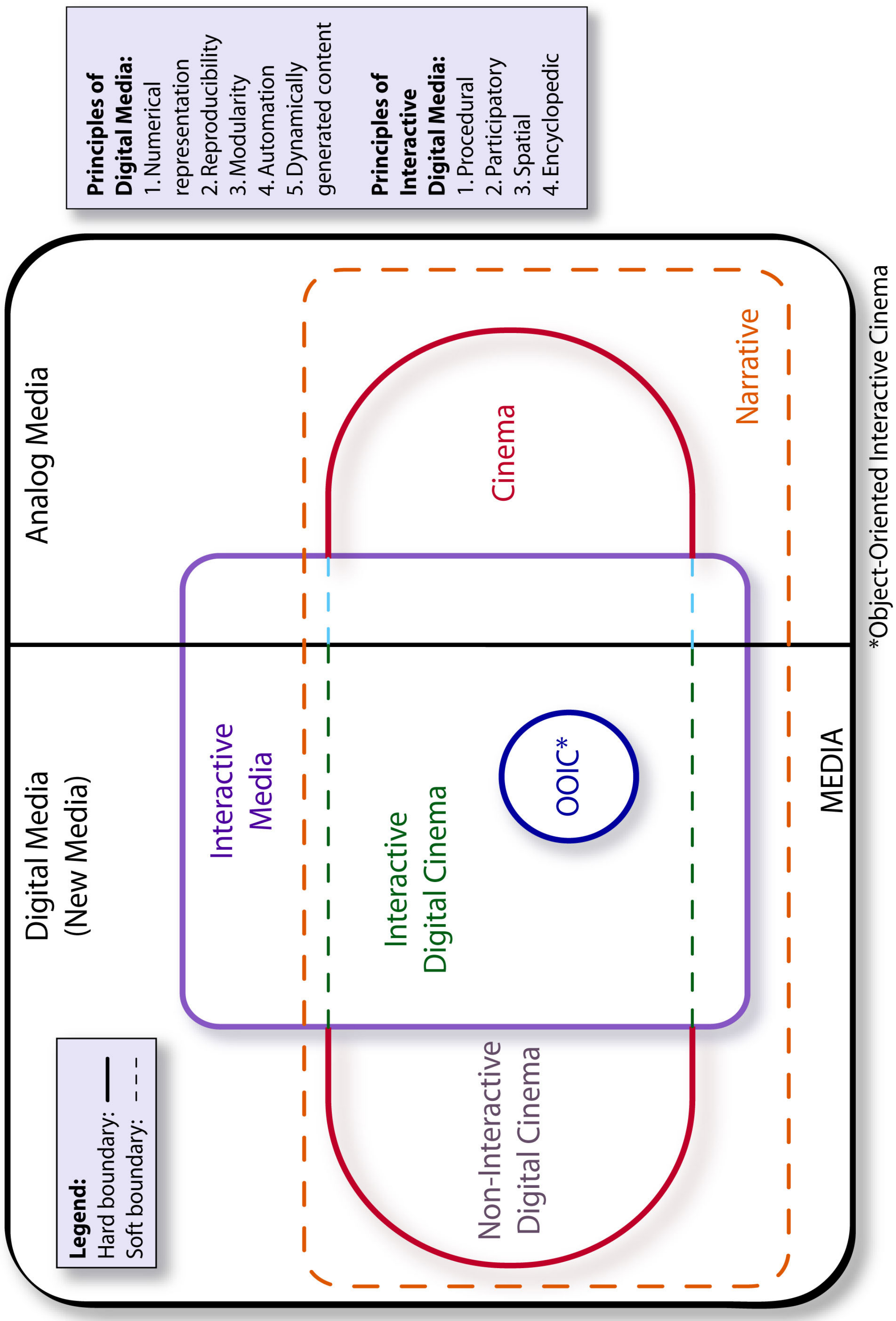


**Figure 41:** A schema for the media spheres related to OOIC

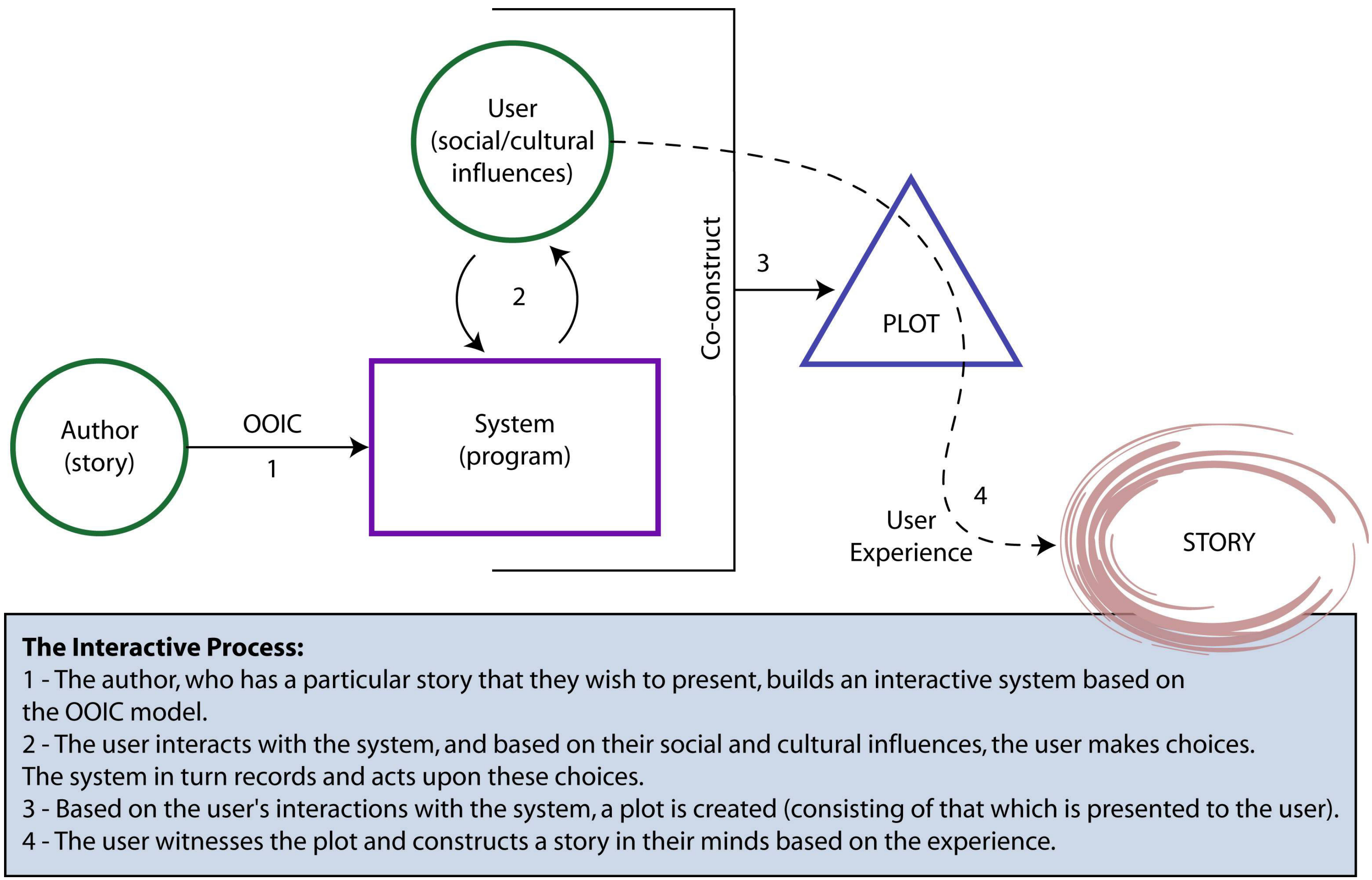


**Figure 42:** The interactive process from author to user-created story

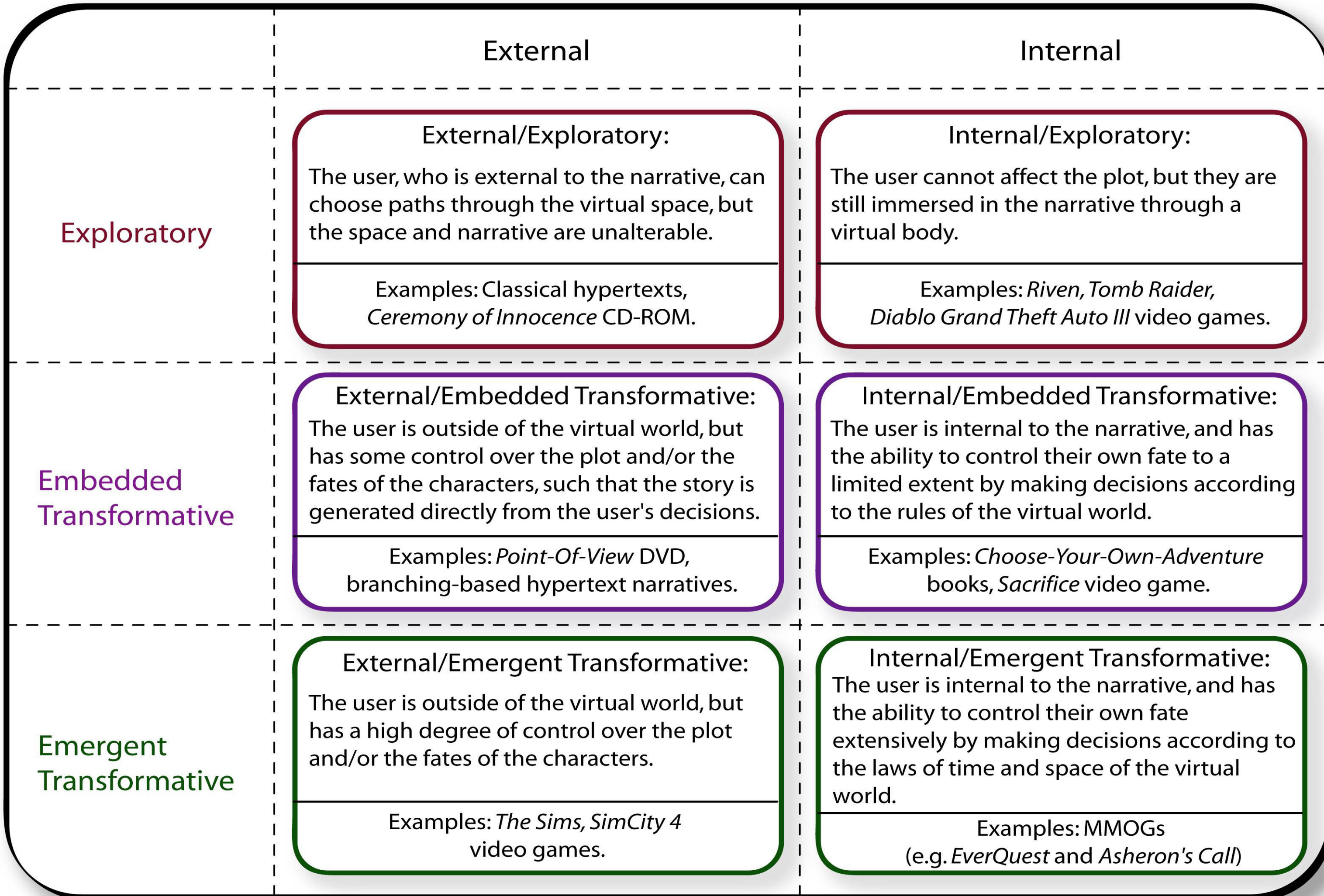


**Figure 43:** A representational model of the six types of interactivity under OOIC

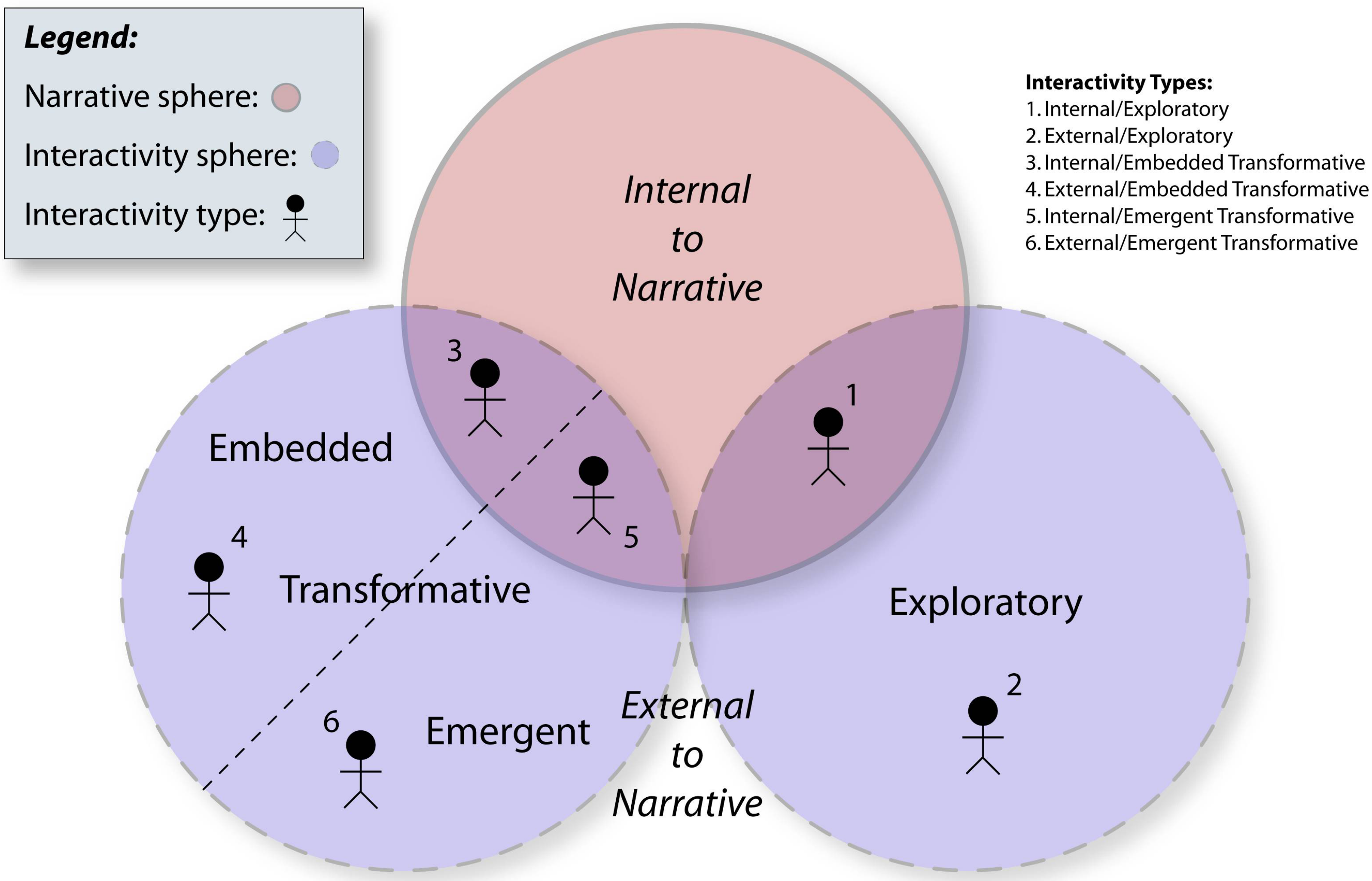


**Figure 44:** A representational model of the OOIC spheres of interactivity

## References


Aarseth, Espen. *Cybertext: Perspectives on Ergodic Literature*. Baltimore, MD: John Hopkins University Press, 1997.

Barthes, Roland. "The Death of the Author" in *Image, Music, Text*. Ed. and trans. Stephen Heath. New York: Hill, 1977.

Beacham, Frank. "Movies of the Future: Storytelling with Computers." *American Cinematographer* April 1995: 36-48.

Bizzocchi, Jim. "Size Does Matter: The Emergent Aesthetics of Flat-Screen Display." 2001. http://www.dadaprocessing.org. Accessed Feb. 4th 2003.

Blom, L.A. & L. Tarin Chaplin. "The Experiential Body of Knowledge," in *The Moment of Movement: Dance Improvisation*. London: Dance Books, 1988.

Booch, G. *Object Oriented Design with Applications*. Redwood City, California: Benjamin/Cummings, 1994.

Bordwell, David. *The Cinema of Eisenstein*. Cambridge, Mass: Harvard University Press, 1993.

Bordwell, David and Kristin Thompson. *Film Art: An Introduction*, 5th ed. NY: McGraw-Hill. 1997.

Branigan, Edward. "Narrative Comprehension and Film." *Sightlines.* Ed. E. Buscombe. Vol. 1. NY: Routledge, 1992.

Brooks, Kevin M. "Do Story Agents Use Rocking Chairs? The Theory and Implementation of One Model for Computational Narrative." Boston, MA: ACM Multimedia 96, 1996.

Buhr, R. "Machine charts for visual prototyping in system design." Technical Report 88-2, Carlton University, August 1988.

Carter, Everett F. "Markov chains." 1996. http://www.taygeta.com/rwalks/node7.html. Accessed Jan. 4th 2003.

Chatman, Seymour. *Reading Narrative Fiction*. New York: Macmillan Publishing Company, 1993.

Cope, David. *Experiments in Musical Intelligence*. Madison, Wisconsin: A-R Editions, Inc., 1996.

Cope, David. *The Algorithmic Composer*. Madison, Wisconsin: A-R Editions, Inc., 2000.

Costikyan, Greg. "Where Stories End and Games Begin." http://www.costik.com/gamnstry.html. Accessed April 5th 2003.

Crawford, Chris. "Understanding Interactivity." Self-published manuscript, 2000.

Crawford, Chris. "Interactive Storytelling Technology." http://www.erasmatazz.com/userdocs/Intro.html. Accessed April 4th 2003.

Davenport, Glorianna. "Smarter Tools for Storytelling: Are the just around the corner?" *IEEE Multimedia* Spring 1996: 10-14.

Davenport, Glorianna. "Whose Bits are they Anyway?" *IEEE Multimedia* July-September 1997.

Davenport, Glorianna. "Curious Learning, Cultural Bias, and the Learning Curve." *IEEE Multimedia* April-June 1998.

Davenport, G., Smith, T. A. and N. Pincever. "Cinematic Primitives for Multimedia." *IEEE Computer Graphics & Applications* 1991.

Davenport, G., Evans, R., and M. Halliday. "Orchestrating Digital Micromovies." *Leonardo* v. 26 no. 4 1993: 283-288.

Davenport, G. and M. Murtaugh. "Automatist Storyteller Systems." *IBM Systems Journal* Vol. 36 No. 3 1997.

Davenport, G., Agamanolis, S., Barry, B., Bradley, B. and K. Brooks. "Synergistic storyscapes and constructionist cinematic sharing." *IBM Systems Journal* Vol. 39 Nos. 3&4, 2000.

DeCerteau, Michel. "Reading as Poaching" from *The Practice of Everyday Life* (trans. Steven F. Randall) Berkeley: University of California Press, 1984.

De Champeau, D., Lea, D., and P. Faure. *Object-Oriented System Development.* Addison Wesley, 1993.

Dodge, Charles and Thomas A. Jerse. *Computer Music: Synthesis, Composition, and Performance*, 2nd ed. New York: Schirmer Books, 1997.

Eisenstein, Sergei. *Film Form: Essays in Film Theory*. San Diego, CA: Harvest Books, 1969.

Epp, Susanna S. *Discrete Mathematics with Applications,* 2nd ed. Boston: Brooks/Cole Publishing Company, 1995.

Felluga, Dino. "Introduction to Narratology." http://www.sla.purdue.edu/academic/engl/theory/narratology/index.html. Accessed March 28th 2003.

Forster, E. M. *Aspects of the Novel*. Orlando, Florida: Harcourt, Inc., 1927.

Gander, Pierce. "Story and Discourse in the Analysis of Narrative." Oct. 1997. http://www.pierregander.com/phd/courses/narratology/culler.html. Accessed March 28th 2003.

Galyean, Tinsley A. *Narrative Guidance of Interactivity*. Ph.D. Thesis, M.I.T., 1995.

Gena, Peter. "Lejaren Hiller (1924-1994)." 1994. http://www.artic.edu/~pgena/lhobit.html. Accessed April 6th 2003.

Graham, Ian. *Object-Oriented Methods*, 2nd ed. Workingham, England: Addison-Wesley Publishers Ltd., 1994.

Hirsh, Haym. "Interactive Fiction." *IEEE Intelligent Systems* Vol. 13 No.5 1998.

Huhtamo, Erkki. "Silicon remembers Ideology, or David Rokeby's meta-interactive art." http://www.interlog.com/~drokeby/erkki.html. Accessed Jan. 22nd 2003.

Kasner, Glenn. "Machine Tongues VIII: The Design of a SmallTalk Music System." *Computer Music Journal* 4 no. 4 1980: 4-14.

Kapor, Mitchell. "The Software Design Manifesto." http://www.kapor.com/homepages/mkapor/Software_Design_Manifesto.html. Accessed Nov. 8th 2002.

Lafia, Marc. "In search of a Poetics of the Spatialization of the Moving Image." *Rhizome Digest*. http://www.rhizome.org. Accessed Jan. 8th 2003.

Landow, George. "Hypertext 2.0: The Convergence of Contemporary Critical Theory and Technology." Baltimore: The John Hopkins UP, 1997.

Lebowitz, Michael. "Creating Characters in a Story-Telling Universe." *Poetics* 13 1984: 171-194.

Levinson, S.C. "Space and Place," in *Some of the Facts* – Exhibition Catalogue for Anthony Gormley at Tate St. Ives, 2001.

Lovell, Robb. "A Blueprint for Using a Reactive Performance Space." http://www.intelligentstage.com/papers/blueprint/Blueprint.html. Accessed Dec. 15th 2002a.

Lovell, Robb. "Video Based Sensing in Reactive Performance Spaces." http://www.intelligentstage.com/papers/VBS.html. Accessed Dec. 15th 2002b.

Lovell, Robb. "Towards Computer Cognition of Gesture." Institute for Studies in the Arts, Arizona State University. http://www.intelligentstage.com/papers/gesture%20paper.html. Accessed Dec. 15th 2002c.

Lovell, Robb. "Computer Intelligence in the Theater." http://www.intelligentstage.com/papers/cit/cit.html. Accessed Dec. 15th 2002d.

Manovich, Lev. *The Language of New Media*. Cambridge, Mass.: MIT Press, 2001.

Murray, Janet. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. Cambridge, Mass.: MIT Press, 2000.

Parisi, Paula. "Grand Illusion: The master of myth rewrites history." *Wired* Issue 7.05 May 1999. http://www.wired.com/wired/archive/7.05/illusion.html. Accessed Feb. 4th 2003.

Pauen, P., Voss, J., and H.-W. Six. "Modeling Hypermedia Applications with HyDev." In Sutcliffe, A. *et al.* (Eds.), *Designing Effective and Usable Multimedia Systems*. Boston: Kluwer, 1998: 23-39.

Petersen, Tracy L. "Interactive Digital Composition." *Proceedings of the 1978 International Computer Music Conference*. Northwestern University, 1978: 167-174.

Preiss, Bruno R. "Data Structures and Algorithms with Object-Oriented Design Patterns in Java." 1998. http://www.brpreiss.com/books/opus5/html. Accessed Feb. 7th 2003.

Purchase, John. "Timecode: Object-oriented movie making?" *Castek Software Factory*, 2000. http://www.cbd-hq.com/PDFs/cbdhq_000601jp_timecode.pdf. Accessed March 1st 2003.

Rees, A. L. *A History of Experimental Film and Video*. London, British Film Institute, 1999.

Robertson, Barbara. "Attack of the Clones." *Computer Graphics World* July 2002. http://cgw.pennnet.com/home.cfm. Accessed Feb. 4th 2003.

Rodet, Xavier, and Pierre Cointe. "FORMES: Composition and Scheduling of Processes." Computer Music Journal 8 no. 3 1984: 32-50.

Rokeby, David. "The Construction of Experience: Interface as Content" in *Digital Illusion: Entertaining the Future with High Technology*, Clark Dodsworth, Jr., Contributing Editor. ACM Press, 1998.

Rokeby, David. "Transforming Mirrors." http://www.interlog.com/~drokeby/mirrorsintro.html. Accessed Dec. 15th 2002.

Rokeby, David. "Interactive Strategies and Dialogical Allegories." http://www.interlog.com/~drokeby/daubner.html. Accessed Jan. 22nd 2003.

Rokeby, David. "David Rokeby: Lecture for 'Info Art,' Kwangju Biennale." http://www.interlog.com/~drokeby/install.html. Accessed Jan. 22nd 2003b.

Rumbaugh, J., Blaha, M., Premeru, W., Eddy, F., and W. Lorensen. *Object-Oriented Modeling and Design*. New Jersey: Prentice-Hall Inc., 1991.

Ryan, Marie-Laure. *Possible Worlds, Artificial Intelligence, and Narrative Theory*. Bloomington and Indianapolis: Indiana University Press, 1991.

Ryan, Marie-Laure. "Beyond Myth and Metaphor: The case of narrative in digital media." *International journal of computer game research* Issue 1 July 2001.

Sawhney, N., Balcom, D. and I. Smith. "HyperCafe: Narrative and Aesthetic Properties of Hypervideo." Hypertext '96, Washington DC, 1996.

Skov, M. B., and J. Stage. "Designing Interactive Narrative Systems: Is Object-Orientation Useful?" *Computers & Graphics* Vol. 26 No. 1 2002: 57 – 66.

Stern, Andrew. "Deeper Conversations with Interactive Art, or Why Artists Must Program." *Convergence: The Journal of Research into New Media Technologies* Vol. 7 No. 1 Spring 2001.

Strohecker, Carol. "A Case Study in Interactive Narrative Design." ACM, 1997.

Tafler, David. "When Analog Cinema Becomes Digital Memory." *Wide Angle* Vol. 21 No. 1 January 1999: 177-200.

Tobias, Ronald B. *Twenty Master Plots (And How to Build Them)*. Cincinnati, OH: Writer's Digest Books, 1993.

Weilberg, Birk. "Beyond Interactive Cinema." http://www.keyframe.org/txt/interact. Accessed Dec. 15th 2002.

Weinbren, Grahame. "In the Oceans of Streams of Story." Millenium Film Journal No.28, Spring 1995.

Wimberley, Darryl and Jon Samsel. *Interactive Writer's Handbook*. San Francisco: Carronade Group, 1996.

Wren, C. R., Azarbayejani, A., Darrell, T., and A. Pentland. "Pfinder: Real-Time Tracking of the Human Body." *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 19 No. 7 July 1997: 780-785.

Zimmerman, Eric. "Against Hypertext." *American Letters & Commentary* Issue #12 2001.