Title
Authorial Leverage: Artificial Intelligence for Narrative and Storytelling

Author
Chen, Sherol

Publication Date
2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

AUTHORIAL LEVERAGE: ARTIFICIAL INTELLIGENCE FOR NARRATIVE AND STORYTELLING

A dissertation submitted in partial satisfaction
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Sherol Chen

March 2021

The Dissertation of Sherol Chen is
approved:

Professor Michael Mateas, chair

Professor Noah Wardrip-Fruin

Professor Warren Sack

Quentin Williams
Acting Vice Provost and Dean of Graduate Studies

Table of Contents

Chapter 1 – Introduction 1

1.1 Traditions – Intended Audience 3

1.1.1 Game and Narrative Designers 3

1.1.2 Game Studies Community 3

1.1.3 Intelligent Narrative Technology Researchers 4

1.2 Contributions 4

1.3 Terminology Disambiguation 8

Chapter 2 – Problem Background – A Case for Variations 15

2.1 Why Narratology? – A Vocabulary for Variations 17

2.2 The Player's Plot – Meaningful Variations 19

2.3 Referable Content – Managing Variations 23

2.4 Satellites and Kernels – Authoring Variations 26

Chapter 3 – Linear Storytelling Background 33

3.1 Storytelling as Authoring Events 33

3.2 Storytelling as Authoring Variations 36

Chapter 4 – Interactive Storytelling Background 41

4.1 Interactive Storytelling as Modes of Play 41

4.2 Interactive Storytelling as Meaningful Variations 45

Chapter 5 – Storytelling AI Background 51

5.1 Intelligent Narratives in terms of what we've discussed so far 52

5.2 Early AI and Believability 55

5.3 Chatbots: Modeling Reactive Believability 57

5.4 Goldwater Machine: Modeling Systems of Belief. 59

5.5 Terminal Time: Combining Reactive Believability with Models of Belief 64

Chapter 6 – Declarative Optimization Based Drama Management (DODM)69

6.1 Drama Management System – EMPath 73

6.2 User Evaluation for DODM 76

6.3 Designing AI Systems for the Author 77

Chapter 7 – Authorial Leverage 81

7.1 Evaluating DM via equivalents 84

7.1.1 Decision Trees 86

7.1.2 DM policy evaluations in EMPath 87

7.1.3 Ease of Expanding the EMPath world 92

7.1.4 Variability of Stories for EMPath 96

7.2 Decision-tree policy issues 97

7.3 Authorial Leverage Informed Storytelling 98

Chapter 8 – RoleModel Background 102

8.1 RoleModel System 103

8.1.1 Authorial Leverage 111

8.2 Rhetorical Variations 114

8.2.1 Hot Cognition 114

8.2.2 Terminal Time 115

8.2.3 Story Skeletons 115

8.3 Logic 117

8.3.1 Constraint Satisfaction 117

8.3.2 Logic in Storytelling 118

Chapter 9 – RoleModel Study - ROBOT DETECTIVE **120**

9.1 Rashomon Effect **120**

9.2 RoleModel + Robot Detective **121**

9.2.1 The Authoring Process 122

9.3 System Design **123**

9.3.1 Objects 125

9.3.2 Verbs 127

9.3.3 Fluents 127

9.3.4 Story Skeletons 129

9.3.5 Rhetorical Parameters 131

9.3.6 Presentation 133

Chapter 10 – RoleModel Results **137**

10.1 DISCUSSION **139**

Chapter 11 – Conclusion **142**

11.1 Future work **143**

11.1.1 Game and Narrative Design 143

11.1.2 Game Studies 143

11.1.3 Intelligent Narrative Technologies 144

Chapter 12 – REFERENCES **145**

List of Figures

Figure 1-1. General breakdown of Authorial Leverage. 5

Figure 1-2. Non-Interactive Story from Author to Audience. 6

Figure 1-3. Interactive Story from Author to Audience.. 6

Figure 2-1. Layers of Interactive Storytelling. 21

Figure 2-2. Operationalizing the Story/Discourse dichotomy. 23

Figure 2-3. Referable content as stories progress. 24

Figure 2-4. Representing Constituent and Supplementary story elements in Façade...27

Figure 2-5. Operationalizing the Constituent /Supplementary dichotomy............27

Figure 2-6. Theoretical breakdown from the author, system, and user's point of views. 28

Figure 2-7. An overall distinction of story and discourse...29

Figure 2-8. Applying narrative terminology to variations. 30

Figure 2-9. Story Experiences: interactive versus intelligent.....................................31

Figure 3-1. Formal breakdown of narrative. (Chatman, 1999)...................................34

Figure 3-2. The framework for narrative that we will be using for this work 35

Figure 3-3. Variations exist in both the interpretations and the choices of the audience...39

Figure 4-1. Methods of Interactive Storytelling (Crawford) 45

Figure 4-2. Final Fantasy 13’s linear structure. 46

Figure 4-3. Branching bottleneck...46

Figure 4-4. Foldback 48

Figure 4-5. Different outcomes for the conclusion of Heavy Rain. 48

Figure 4-6. Reducing branching factor, by starting with foldback and introducing branching at the very end. 49

Figure 5-1. Indirect relationship between interactive and intelligent stories. 52

Figure 5-2. Final Fantasy 1, 2, 7, and 13 evolution of character graphics. 54

Figure 5-3. Evolution of Pikachu throughout the Pokemon series............................54

Figure 5-4. Conversation with chatbot, Eliza. 58

Figure 5-5. Conversation between two chatbots, ELIZA and PARRY. 59

Figure 5-6. Abelson's Rationalization Mechanisms from his paper on Hot Cognition. 62

Figure 5-7. Terminal Time System Diagram. 66

Figure 5-8. Timeline of content and audience interaction. 67

Figure 6-1. Drama Manager Architecture...70

Figure 6-2. Plot points for the EMPath DODM System...71

Figure 6-3. Drama Manager Components...73

Figure 6-4. EMPath 1.0 World Map. 74

Figure 6-5. When a drama manager action is triggered, the game world communicates it through story events, causing a plot point..............................75

Figure 7-1. This is the directed-acyclic graph for the story world. 88

Figure 7-2. Histogram of story qualities given the search policy...............................89

Figure 7-3. The poorly evaluating decision 1000-tree (17 nodes)...............................89

Figure 7-4. The highest evaluating decision 200-tree (70 nodes)...............................90

Figure 7-5. Part of a decision tree built from stories generated by DODM. Zoomed in view of the 200 pruned tree. 91

Figure 7-6. Approximated complexity for the most optimal decision tree policy. 93

Figure 7-7. Histogram for unique stories according to tree size. 97

Figure 8-1. RoleModel input and output pipeline. 104

Figure 8-2. RoleModel System Architecture. 107

Figure 8-3. HTML visualization of story answer set role assignment 108

Figure 8-4. Example code. 109

Figure 8-5. Roger Schank's example story skeletons. 117

Figure 9-1. The fixed image assets described in Table 9-1. 123

Figure 9-2. A representation of ROBOT DETECTIVE as distinct rulesets. 124

Figure 9-3. Physical objects. 126

Figure 9-4. Fluents. 128

Figure 9-5. From logical output to presentation. 134

Figure 9-6. Story output with image and natural language interpretation 135

Figure 9-7. Example story output. 135

List of Tables

Table 1-1. What is managed when we tell stories. 7

Table 1-2. Narrative formalisms that help create the building blocks for this work. 10

Table 1-3. The audience experiences events while the author experiences variations. The audience views the world as story and discourse, while the author views as constituent and supplementary. 12

Table 2-1. Three frameworks for storytelling 17

Table 2-2. An in-general overview of storytelling models. 32

Table 4-1. Modes of Play 43

Table 5-1. Computers taking on human aspects of storytelling. 53

Table 5-2. Representations of chapters 3 and 4 as it relates to Intelligent Narratives (chapter 5)..55

Table 5-3. Example list of Intelligent Systems..59

Table 7-1. Authorial Leverage 83

Table 7-2. Game policy variations...92

Table 8-1. General definitions for Authorial Leverage 111

Table 8-2. An in-general overview of storytelling models. 112

Table 8-3. Comparing our new approach, Declarative-Rhetorical, with an older approach.. 113

Table 8-4. Abelson's definitions for framing human affect as computational problem-solving. 114

Table 8-5. Abelson's mechanisms for maintaining human bias. 115

Table 9-1. The describing the fixed image assets of ROBOT DETECTIVE. 123

Table 9-2. Overview of the major components of ROBOT DETECTIVE................. 125

Table 9-3. Verbs. 127

Table 9-4. Fluent types. 128

Table 9-5. Story Skeletons and their meaning. 129

Table 9-6. Rhetorical parameters, represented by role assignments, and their meaning. 132

Table 9-7. Role Nullifiers inspired by Abelson's rationalization mechanisms and Schank's story skeletons. 133

Table 9-8. Generative storytelling paradigms listed on the left with RoleModel's approach on the right 136

Table 10-1. The image captions for Trial A and Trial B. 138

Table 10-2. Results of the ROBOT DETECTIVE study. 138

Abstract

Sherol Chen

Authorial Leverage: Artificial Intelligence for Narrative and Storytelling

Intelligent Narrative Technologies have changed storytelling by facilitating new types of story experiences, such as interactive stories. However, Intelligent Narrative Technologies also introduce a prohibitively high cost of authorship, which goes beyond the effort it takes to learn how to use new technology and tools. New technology replaces aspects of traditional human authoring with instructions and procedures. Evaluating the power of expressiveness in different storytelling approaches is similar to understanding the difference in utility between a hammer and a nail gun, or between Python and Assembly.

This dissertation asks why such authorship is so difficult, and proposes a technique for making it easier. The three main contributions of this work are: (1) it provides an author-centric design model for intelligent narrative, positioning storytelling as the management of variations; (2) it introduces the Authorial Leverage (AL) framework, a means for evaluating the costs and benefits of interactive story authorship; and (3) it introduces a system designed to demonstrate the aforementioned concepts.

The author-centric design model sees interactive stories as composed of constituent and supplementary events, as opposed to the traditional story-and-discourse model. This allows us to build interactive storytelling tools with greater authorial leverage. We describe one such system, called RoleModel.

Dedication

At a very young age, video games taught me that perseverance was a way to escape lesser circumstances.

In video games, the main character often faces some of the most tragic beginnings. In fact, some of the most hopeless situations start protagonists on their Hero's Journey to right the wrongs of the world. And as long as they didn't quit, they could fail a million times knowing that a better world was possible.

When I was 10, I had lost all my hope and, in a moment of asking why life is worth living, I had one of my most vivid experiences to date. What I learned was:

1. That people weren't happy because they lived primarily for themselves. This is not what we are meant for. Helping even five people in my lifetime would be worthy of living.
2. There's a message I carry that no one who has lived before or after me will have, and the world would miss out if I didn't exist. That the canvas for this message may not even exist yet, and I ought not to try to understand my message before it's ready. This is true for every person.
3. Finally, that there will be others when I get there.

I still don't understand, but I keep finding myself in places I'd never imagined I'd go. While I still feel like a foreigner in most circles. I've taken stock in learning, adapting, and persevering, forgiving myself and anyone else along the way.

I don't know how or when I will get there, but regardless, I dedicate this and all my work to God for giving me my purpose and the will to overcome all the challenges/failures life brings and to keep trying for the best possible world in my lifetime.

Acknowledgements

Professional Acknowledgements:

Thanks to my committee: Michael Mateas, Noah Wardrip-Fruin, Warren Sack

Thanks to my colleagues, classmates, and collaborators: Aaron Reed, Adam Smith, Andrew Duensing, Anne Sullivan, Arnav Jala, Ben Samuel, Chris Lewis, David Olsen, David Thue, James Skorupski, Josh McCoy, Kate Compton, Ken Hullett, Mark J. Nelson, Mark Riedl, Mike Treanor, Mirjam Eladhari, Peter Kong, Peter Mawhorter, R Michael Young, and many others

Thanks to the friends who helped read, edit, and analyze data for this dissertation: Emily Moberg Robinson, Jon Liang, Chaim Gingold, Ryan Proch, and Tracy Holsclaw

Thanks to my mentors who immovably encouraged me to finish and submit this dissertation: Doug Eck, Heidi Baker, Jacquelline Fuller, Noah Falstein, Glen Davis, Rick Sommer, Stephan Bohacek, and Maria Palacas

Thanks to the Expressive Intelligence Studio, University of California at Santa Cruz and the National Science Foundation

Personal Acknowledgements:

A general thanks to my parents, family, and friends.

Thanks to the families that adopted me, taking me in for the holidays, and offerred spiritual support: Eileen and Rod Williams, Janet and Peter Payne, John Rodriguez, Karen and James Jurado, Kathy Thill, Kristina and Brandon Johnson, Laura and Dave Gschwend, Leigh Durst, Pickens Family, Rachel and Donny Greer, Shauna and Ken Cornell, Sherrie and Jim Grabot, Susan and Keith Schaad, Tracy and Greg Holsclaw, Trudi Iverson, and Wendy and Keith Lindstrom, and others

Thanks to my PhD musician friends from the Terminal Degree Jazz Band community: Andrew Short, Ben Akiyama, Casey Matteson, Chandra Rajagopal, Chris Davis, Chris Motter, Cory Graves, Daniel Brown, David Fryauf, David Goodman, Derek Mendez, Devon Wyland, Dominic Calhoun, Dotan Negrin, Emil Carson, Isaiah Roberts, Jason Lee, Jensen Tan, Jewl Sandoval, John Dalessi, Jordan Johnson, Josh Brahinsky, Kenn Knowles, Kurt Stockdale, Lex Olsen, Matz Haugen, Olivia Frise, Rachel Steinhart, Raj Maitra, Rosemarie Echelson, Sarah Pickens, Sebastian Wolff, Wes Souza, Yu Yahagi, Yves Tan, and many others

Thanks to those who impacted me in different points of my life, those who've inspired me, and those who believed in me (not mentioned above):

Career – Anna Huang, Carter Morgan, Colt McAnlis, Daphne Ippolito, Daphne Luong, David So, David Ross, Deanna Chen, Erin Hoffman-John, Greg Wilson, Joanna Ng, Justin Zhao, Kelsey Hightower, Lucy-Zayne Nsour, Maya Chatav, Michal Todorovic, Natasha Jaques, Nina and Bob Liang, Valentina Nesci, Wes Cheng, and others

Graduate School – Alice Ye, Alicia Haley, Bala Rajaratnam, Brenda Romero, Carol Reiley-Ng, Craig Reynolds, Damian Isla, Daniel Kline, Diane Lee, James Corby, James Davis, Jamil Moledina, Jim Grove, Lucinda Robledo, Paul Contos, Richard Evans, Rob Klevin, Taylor Barrella, Tracie Tucker, Veronica Larkin, and others

Undergraduate - Clif Johnson, David Kovara, David Sauders, Errol Llyod, Fred Adams, Gary Zoppetti, Harvey Price, Heyward Brock, Jim Ancona, Jim Tweedy, John Courtright, Karl Unruh, Keith Decker, Kevin Ettiene Cummings, Mary Martin, Michael Arenson, Michelle Fillings Brown, Phil Conrad, Terry Harvey, Todd Groves, Tom Palmer, Vernon James, and others

High School - Brent Thorpe, Brian Casey, Chrystal Haas, Elliot Seifert, Gus Highfield, Joe Henderson, Kelly Kline, Mrs. & Mr. David Byers, Ms. Murphy, Olga Knickerson, Peter Parlett, Richard Liu, Robert Teate, and others

Spiritual Leadership – Al Erisman, Bruce Cooke, Carolyn Martin, Dan Kimball, Dave Chae, Dave Gibbons, Angie and Dave Blackwell, Doug Perkins, Grace Samson-Song, Gina

Hyatt, Ken Williams, Kyla Craig, Michael McNally, Mrs. and Mr. Knettler, Nancy Duarte, Nancy Ortberg, Roz Picard, Susie Lipps, and others

Passion Talks Advisors: Danny Kim, Holly Liu, John P. Shen, Nicole Dickens, Rob Pace, Ron Hicks, Vip Patel, and others

PhD Pandemic Friends - Bret Staudt Willet, Hannah Eagleson, Jeff and Annabelle Ho, Julia Choi, Kimmy Wu, Lulu Chen, Marquise McGraw, Nadia Benbernou, Richard Zhang, Tammy Chang, Will Hart

Other Groups – Academic Bridges, All Nations Church, Delaware Allstate Choir, Faith & Work Leaders, Gathering by the Bay/Firestarters, GDC CA Program, Global Tech Forerunners, Golden Gate Radio Jazz Orchestra, Google Brain, Google Cloud, Google Christian Fellowship, Google Christians in AI, Google Interactive/Games, Google Machine Learning Ninjas, McKean Marching Band/Percussion, McKean Track & Field, McNair Scholars Program, Monterey Jazz Festival, Passion Talks Community, Progressive Baptist Gospel Choir, Project Magenta, Stadia *Lab, Santa Clara Vanguard Big Band, Stanford Chi Alpha, Stanford Pre-Collegiate Studies, UCSC Department of Computer Science, UCSC Graduate Christian Fellowship, UCSC Graduate Student Housing, UCSC Jazz Band, UCSC Residential Life, UCSC School of Engineering, UDel Blue Hen Ambassadors, UDel Christians & Prayer Nights, UDel Computer Science, UDel Gospel Choir, UDel Honors Program, UDel Jazz Band, UDel Marching Band/Percussion, UDel Music, UDel Philosophy, UDel Residential LIfe, UDel TRiO, Vintage Faith Community Group, YouTube Radio, and many others

Chapter 1 – Introduction

> From an instant to eternity, from the intracranial to the intergalactic, the life story of each and every character offers encyclopedic possibilities. The mark of a master is to select only a few moments but give us a lifetime (McKee, 1997). - Robert McKee, creative writing expert, describes what it means to tell stories.

When you pick up the controller, you are engaging in the goals and objectives of the world presented to you. Like in real life, you've become a determining factor in the fate of this world. Of course you must (eventually) succeed, in developing the skills to achieve the objectives for the greater good. Common questions game developers think about is how clear this divide is between good and evil, whether this distinction is even important, or whether this is even the right distinction. As these objectives are set, who decides what they actually mean— the author or the player? As we formalize this process (for machines), these questions become less trivial in the design process for stories in interactive spaces. This work provides insight for how we teach computers to tell stories.

To introduce this work, we set the stage for the sort of storytelling we are interested in. While the theories and studies discussed in this dissertation apply to all of storytelling, this work examines storytelling in digital spaces, video games, and interactive and generative stories. In particular, it sheds light on how new forms of storytelling tools can effectively impact the authoring experience. Technology and tools with more Artificial Intelligence capabilities currently are being developed that offload the authoring practice to machines, lighten the authoring burdens, and create newer types of (more dynamic) audience experiences.

How can we tell stories more effectively? There are expert storytellers, people who study storytelling, and people who build tools for storytelling. As we develop better technology and tools, we deepen our understanding of how we tell stories. Correspondingly, having a better understanding of how stories are told helps us build better technology and tools. For example, Intelligent Narrative Technologies enable us not only to grow our understanding of the storytelling process, but also to create new ways of telling stories. We want to identify where we can improve our digital storytelling, why it's challenging, and the ways to break down the problem to make it easier.

First, we look at the previous work on and the evolution of the problem space. We want to create better story experiences, but we also want to be able to explain what "better" means. In any story exchange, there is an author and an audience. As we develop more tools, the authoring process becomes more complex. Although there is a complementary relationship between audience and author, this work mainly explores the complexity of authoring stories in digital spaces.

1.1 Traditions – Intended Audience

In this section, we discuss our audience and the communities we intend to impact. In particular, this dissertation engages the fields of Game and Narrative Design, Game Studies, and Intelligent Narrative Technologies. We also lay out how our work impacts these communities.

1.1.1 Game and Narrative Designers

A primary motivation for this work is to benefit the community of Game Developers, whether independent, commercial, or artistic. Traditionally, the story in games often comes in conflict with giving the user opportunities to have dramatically significant impact (Chen, 2009). This work explores ways in which improving user experience does not come at the cost of unreasonable increase in authorial burden. As technology enables us to tell better stories, new areas of expertise, such as narrative engineer or narrative AI engineer will become essential to game development.

1.1.2 Game Studies Community

Theorists like Janet Murray, Brenda Laurel, and Marie Laure-Ryan use the fundamentals of narratology and drama to explain the structure of interactive and digital narratives (Murray, 1997; Laurel, 2013; Ryan 2006). In looking at the authorial burdens associated with bringing computers into the storytelling process, this work moves from an audience-centric to author-centric emphasis in story design. Although not the sole focus of the Game Studies community, narrative and storytelling currently are major areas of discussion. This work continues the conversation about how to bridge narratology and the critical study of games.

1.1.3 Intelligent Narrative Technology Researchers

Michael Mateas proposed using computers to manage story experiences in order to achieve higher player agency and dramatic story significance (Mateas, 2001). The Intelligent Narrative Technology community is the community of AI researchers interested in using technology to expand and enhance storytelling. Using AI in story potentially can create authorial burden in other aspects of the creation process, however. This work defines an Authorial Leverage framework for evaluating the burden on and leverage of AI systems, and evaluates its ability to create compelling story variations with less authorial effort.

1.2 Contributions

This work makes three major contributions. First, it redefines the problem in regards to the author's experience. Second, it analyzes AI systems; in particular, it identifies the leverage that storytelling practices give the author. Traditionally, when we study stories, we look at the end-user experience. However, when telling stories using intelligent narrative technologies, it is also important to consider the author. Finally, it presents a storytelling system, RoleModel, that demonstrates the latter approach.

As mentioned above, there is a complex cost-benefit tradeoff to consider in using AI for communicating and representing stories. AL (Authorial Leverage) is an author-centric evaluation model that compares the costs and benefits of using AI (or any sort of method/approach) to design a story experience. It considers the user experience, but it also aims to evaluate and model the author's experience. We discuss this in more detail in the following chapters.

$$\text{Authorial Leverage} = \frac{\text{Audience Experience}}{\text{Authorial Effort}}$$

Figure 1-1. General breakdown of Authorial Leverage.

The field of Narratology studies stories by breaking them down into two distinct components: the content of possibilities, and the delivery of said content. McKee describes the materials available to the storyteller as "the life story of each and every character, [which] offers encyclopedic possibilities." The narrative, or the practice of the storyteller, aims to pick a few moments to capture a lifetime. Technology, however, redefines the role of the storyteller by obscuring the traditional roles of audience and author. In digital spaces, for example, we can easily design interactive stories that give authorship to the audience. Now, we not only author the story, but we also author the ability for the audience to create their own story. Figure 1-2 Figure 1-3 below give a general sense of how an author delivers a story to their audience. In this work, storytelling is defined as the materials available to be communicated (content) in addition to how it's presented (discourse).

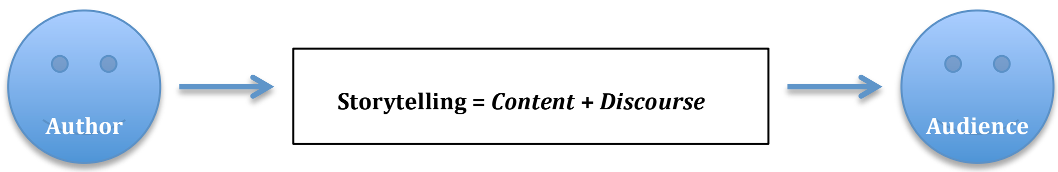

Figure 1-2. Non-Interactive Story from Author to Audience.

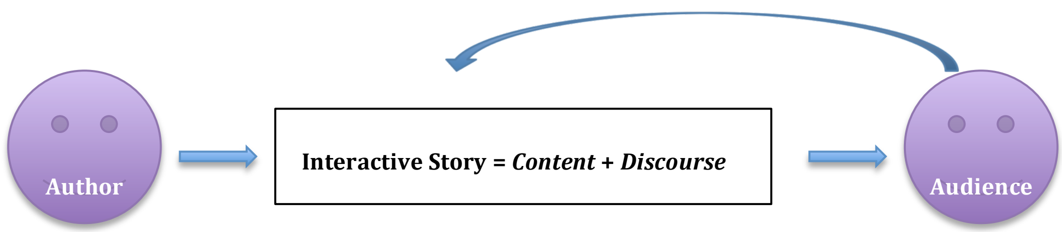

Figure 1-3. Interactive Story from Author to Audience.

Even in ostensibly non-interactive pieces, audiences "rewrite" stories with their interpretations and biases. For example, suppose my relative has stage 4 cancer, receives prayer, and is determined by medical evaluation to be cancer-free the next day. This faith-healing narrative translates differently to different people. For some, the prayer is a prime mover in my relative's recovery, for others, it's just a coincidence.

As Abbott says, "We don't really believe something is true unless we can see it as a story." (Abbott, 2002) Artificial Intelligence is motivated by offloading human practices to technology (such as computer vision, natural language processing, and robotics). For our work, we are interested in the intelligence involved in how we tell stories. A solid design of the storytelling process becomes more necessary in digital spaces, because, unlike movies and novels, we aren't just designing single stories but rather potential story spaces. We will show (through AL) that a good story design does not give as much leverage as managing story variations of the storytelling process. Variations, therefore, become the primary building blocks for redefining the problem (explored more in Chapter 2).

If authoring stories can translate into managing variations, we shift how we think about storytelling; instead of focusing on the end result, we focus on the process of arriving at the end result. When we consider intelligence, whether natural human behavior or Intelligent Narrative Technologies, we create a deeper understanding of how to better tell stories, thus gaining Authorial Leverage. Rather than working toward better story outcomes by focusing on what there is to tell and how we tell it, we instead manage the space of desired story variations. The process then becomes driven by meaning, belief-systems, and rhetoric, rather than by simply managing events and their delivery. Table 1-1 below gives examples of what we mean when we offer a new approach to telling stories.

Audience-Centric (Events)	Author-Centric (Variations)
Managing story & discourse	Managing variations of story & discourse
Managing the story	Managing story spaces
Managing outcomes of story	Managing authorship and agency within story
Managing what there is to tell & how we tell it	Managing meaning, belief-systems, & rhetoric
Managing the end result	Managing the process
Designing Stories	**Designing Storytelling**

Table 1-1. What is managed when we tell stories.

1.3 Terminology Disambiguation

This work analyzes storytelling, interactive stories, and how technology facilitates authorship. However, the term "story" can still be quite ambiguous, as it is used and defined in various ways throughout narratology and game studies. In the early 1900s, Russian Formalists used the terms fabula, the raw material of the story, and syuzhet, the organization of these materials, to define "story." We use this same dichotomy in this work, as identified in Figure 1-3 above — what we call content and discourse. "Story" later became used to distinguish between the chronological ordering of story content versus the order of telling, or plot, as dictated by the narrator or storyteller. It follows that the telling of the story or narrative is constituted by story and plot. So on the one hand, we have content, fabula, and story; on the other hand, we have discourse, syuzhet, and plot.

For this work, we only need to make the fundamental distinction between the content of story and its discourse. In other words, we won't go into the nuances of story and plot, because structurally, we need at most just enough formalism to design and evaluate AI (storytelling) systems. This is particularly important in games, as player interactions alter either the story (content) itself or the discourse of the story (or both). Marie Laure-Ryan denotes this distinction through what she calls ontological and exploratory interactions. Ontological interactions alter the fate of the world, while exploratory interactions do not (Ryan, 2006). For example, I have never seen a Star Wars movie, and while I do have agency over the order in which I watch the movies, I have no effect on the fate of the Star Wars universe. If we consider a game like Final Fantasy, I can often change the names of a character, which holds no bearing to the fate of the world. In games, we can alter either the story itself or how it is presented to us. This content/discourse distinction covers the type of experiences that the audience or players of the game can have.

The creation of a story or game itself, however, is about managing the space of choices that a player has rather than just managing which choices to make. Since each choice creates a variation in the event of the story, it is better to see stories as made up of the dichotomy between constituent and supplementary events. (Chatman, 1978). In Table 1-2 below, we define what we mean by these terms. We also delineate between the audience's experience and the author's experience as building events versus building variations (shown by the two columns in Table 1-2 below). In the later chapters and as a primary contribution, we argue that supplementary variations improve author agency in generative story systems.

	Events (Audience Experience)	Variations (Author Experience)
Content	The facts and events available	(**Ontological)** The space of outcomes and possibilities that alter the fate of the world
Discourse	The organization and delivery	**(Exploratory)** The different possible ways to tell and deliver the story without altering the fate of the world
Constituent	Events that move a story forward	Variations that are integral to moving the story forward. Changes that can **significantly** alter the fate of the world.
Supplementary	Events that won't move a story	Variations that do not **significantly** alter the fate of the world.

Table 1-2. Narrative formalisms that help create the building blocks for this work.

Traditionally, interactive spaces aim for a greater ontological mode of interaction as the holy grail for interactive storytelling (Ryan, 2001). This adds complexity to the formalisms in narratology, because "story," as it is traditionally used, refers to fixed events set in chronological time. For example, "Little Red Riding Hood" is a story that has been told countless times; therefore, it has many instances of discourse variation. However, you could not say that you don't find "story" variations within the space of Little Red Riding Hood tellings. It's more accurate to view these story variations as insignificant or not so meaningful, since if they were significant, we would find ourselves further outside of the Little Red Riding Hood world. Advances in interactive storytelling, however, aim to create these significant story alterations. It requires a nontrivial paradigm shift to argue that we also must (significantly) compose the "story" (not just the telling), so instead of calling this "story," we use the terms "story spaces" or story "content." We intend to create spaces for stories to compose themselves.

Furthermore, the ontological (altering the fate of the world) and exploratory (non-altering) modes of play reveal how traditional narrative formalisms break down in interactive stories. For example, suppose Little Red Riding Hood brings her grandmother a peanut butter sandwich in one telling, and a turkey sandwich in another. While both variations alter the fate of the world, they do not do so in a meaningful way. A more meaningful insight is that it probably doesn't matter what Little Red Riding Hood brings to her grandmother. The real issue is whether changing the sandwich is discourse-level or not. If there truly existed a Red Riding Hood in real life and she indeed carried a turkey sandwich with her, then saying that she brought peanut butter does change the story. In contrast, not mentioning what she brought may leave the audience unaware of the turkey sandwich, but it wouldn't mislead them either. In other words, omission is a discourse-level, or exploratory, variation, while reporting something contrary to what actually happened is an ontological variation. Altering the fate of the world, therefore, is not enough of a definition around which to fully (and meaningfully) design story variations, as what we really care about are significant alterations to the fate of our story world (and not all alterations are equally meaningful), where significance depends on reader/player background knowledge, interpretation and bias.

Understanding storytelling as story materials and their presentation (content + discourse) is a useful starting point, but, as defined by Table 1-2 and explained in the previous paragraph, it is more useful to view variations in terms of significance to the fate of the story world rather than just whether it is an element of the story itself or its presentation. In Table 1-3 below, we show how the audience experiences stories as events, and how authors experience the storytelling process through variations. A similar divide is made between the content/discourse and constituent/supplementary dichotomies. From the audience's view, there is a fixed story and its delivery, while for the author, it's more accurate to view storytelling as division between events that matter and events that don't. Table 1-3 below shows that the author **experiences** (right column) stories through variations and **views** (bottom row) the creation of story in constituent and supplementary events and variations, while the audience experiences (left column) stories in events, and views (top row) the story through material and discourse events as a observer, and through variations as an active participant.

	Audience Experience	**Authorial Experience**
Audience View	Content (Story) Events Discourse Events	Content (Ontological) Variations Discourse (Exploratory) Variations
Authorial View	Constituent Events Supplementary Events	Constituent Variations Supplementary Variations

Table 1-3. The audience experiences events while the author experiences variations. The audience views the world as story and discourse, while the author views as constituent and supplementary.

In addition to making a content/discourse (or ontological/exploratory) distinction, Chatman makes another distinction: between satellite and kernels.

Abbot designates these as constituent and supplementary events (Abbott, 2002). Constituent events drive the story forward; supplementary events do not. This constituent and supplementary distinction focuses the design of a story around the author's agency in working with technology (or AI) to build a story world. The material/discourse distinction caters to user experiences. Although the material/discourse dichotomy is important in designing stories, it should not be the only means of managing variations. The constituent and supplementary view of storytelling gives the author the authority to maintain the story's integrality, rather than solely relying on the structural content and discourse.

To test our hypothesis, we design a storytelling system called RoleModel (discussed in Chapter 8, Chapter 9, and Chapter 10). While other studies have attempted to address the problem of having computers generate better stories, their approaches often increased authorial burden in other ways. RoleModel creates meaningful variations in stories by leveraging more than just the structural content and discourse. This is an early example of what it means to have a more author-centric design of AI storytelling systems that makes use of the constituent and supplementary paradigm.

This chapter introduced the communities that these discussions and contributions impact, concluding with a brief introduction to the three contributions of this work, further explained in Chapters 2, 3, and 4. This dissertation makes three major contributions to the field. First, it provides an analysis of storytelling as the management of variations. Second, it lays out the

Authorial Leverage framework. Finally, it introduces the RoleModel system of design, implementation, and evaluation. Chapters 2 through 5 lay out the first of the three contributions in discussing the shifts in how we define and understand storytelling. Towards the second contribution, Chapters 6 and 7 details the evaluation of expressivity among storytelling approaches with Authorial Leverage. Chapter 8, Chapter 9, and Chapter 10 outline the RoleModel system, an example of what designing storytelling as the management of variations could look like.

In the next chapter, we discuss the problem and theory. If AI enables the computer to offload the authorial burden from human creators, then why aren't there more immersive and interactive forms of stories?

Chapter 2 – Problem Background – A Case for Variations

> You could in fact argue, and people have, that our need for narrative form is so strong that we don't really believe something is true unless we can see it as a story (Abbott, 2002). - H. Porter Abbott, Narratologist, argues that humans are unavoidably storytellers.

In the previous chapter, we introduced the aspects of story relevant to this research. Moving forward, we want to understand the ways that we can improve storytelling. This chapter identifies current setbacks, and proposes that we shift our framework for understanding storytelling to focus on the process rather than the end-result. Specifically, we articulate why it is necessary to shift from the audience experience to the author's experience as we move from traditional stories towards stories that are intrinsically intelligent. The shift from audience to author is fundamentally a shift from event based understanding of story to variations based. Rather than storytelling through establishing outcomes and events, this work shows it is more useful for storytelling to be about establishing spaces for variations.

In general, what problems do we encounter when we're trying to advance storytelling with machines? This chapter will look at past advances and challenges. Although games look and feel very different today than they did twenty years ago, there has been fewer examples innovation in story design and deployment. In pushing the boundaries of storytelling, the interactive drama, Facade, is a well-known generative story experience (Mateas, 2005). Even though the interactive drama Façade does provide a more generative story experience, using much more AI in its delivery than commercial games, in the last decade, Façade-like stories have not been easily reproduced. Overall, this chapter identifies the authorial burden inherent in innovating and advancing storytelling. This addresses the gap in the literature, which traditionally has not placed the author at the center of its analysis of the difficulty towards the design and implementation of story experiences (like Façade).

To better understand why there is a problem, the following chapters represent areas of previous work most relevant to the contributions of this dissertation. First, we discuss how storytelling has been understood by narratologists—for the most commonplace occurrences like in books and film. Second, we discuss how games and other interactive spaces create a different type of storytelling experience as understood by game theorists. Finally, we identify the barriers and propose augmenting our frameworks for storytelling once more as we pursue stories that are intrinsically intelligent. The frameworks are described in Table 2-1 below as for narrative, interactive narrative, and intelligent narrative. The associated dichotomies, described on the right hand side, identify ways that intelligent stories can be operationalized. The aim of this chapter is to establish these types of variations, informed by these frameworks, as our building blocks. Specifically, we set apart constituent and supplementary variations as the most appropriate basis for understanding and measuring success as we define it through Authorial Leverage.

Three Primary Frameworks	**Associated Dichotomy**
(Linear) Narrative/Storytelling	Content and Discourse
Interactive Narrative/Storytelling	Ontological and Exploratory
(Artificially) Intelligent Narrative/Storytelling	Constituent and Supplementary

Table 2-1. Three frameworks for storytelling

2.1 Why Narratology? – A Vocabulary for Variations

In the pursuit of storytelling on digital platforms, technologists look to narratology and literary theory, notably dating back to Russian Formalism. Marc Cavazza, in his review of Narratology for interactive storytelling[1], states that the most cited work in Interactive Storytelling work is by Vladamir Propp, Russian Formalist. (Cavazza & Pizzi, 2006). We explore traditions of narrative formalisms in story generation and drama management. In his narratology paper, Marc Cavazza uses literary theory as a means to formalize storytelling. He begins with Aristotelian dramatic theory, and then describes the tradition literary structuralism by showing how Propp, Greimas, Barthes, and Bremond

[1] Story generators are primarily focused on the user experience, seeking to produce satisfactory stories. The added power provided by AI is the ability to produce many stories, or a space of satisfactory stories, without having to hand-author each one. Story generation and interactive storytelling are academically distinct areas. Interactive storytelling, however, is a simulated-user away from being story generation. We see the relationship between story generation and interactive storytelling if we equate a play trace from an interactive experience as a form of guided story generation.

have informed existing storytelling systems. His paper surveys theories ranging from poetics to post-structuralism, and argues that the formalization and implementation of storytelling systems recreate literary theory constructs. (Cavazza & Pizzi, 2006)

In kind, the previous chapter looked at the distinction between plot and discourse as Russian Formalist concepts of fabula and syuzhet—the raw material of a story versus the way a story is organized. Applying these concepts to computer games, Marie Laure-Ryan describes the interactive[1] experience of influencing plot versus discourse as the ontological mode of play versus the exploratory mode of play. According the Ryan's theory, the content/discourse dichotomy is appropriate in the user-end experience. The author experience, in contrast, is better understood through the constituent and supplementary formalism, derived from Seymour Chatman's theory of kernels and satellites—the essential components of story versus ornamental details.

Structuralism made way for modeling storytelling to the point where a computer could assist, recreate, or understand the authoring process. When introducing his reader-model for story generation, Paul Bailey summarizes previous work in how computer systems are designed to generate stories. Story generation systems are guided by a variety of models: author models, story models, world models, and reader models (Bailey, 1999). These models attempt to represent where a story gets its meaning and direction. They operationalize event plans, character motivations, dramatic arcs, and other strategies aimed at identifying and fitting together fundamental story pieces. These traditions in story generation are descendants of the structuralist content and discourse way of thinking.

Building on these concepts, in our work, we (1) approach generative storytelling[2] as the management of story variations, (2) identify narrative and game studies constructs of events, discourse, as well as ontological, and exploratory as the traditional types of variations and (3) use Façade as an example of a sophisticated story system to show how the new framework of constituent/supplementary variations contributes to the design process and analysis.

2.2 The Player's Plot – Meaningful Variations

In addition to defining a vocabulary for variations through storytelling dichotomies, we also need to understand how meaningful (to the audience or player) one variation is from another. In her analysis of human-computer interactions, Brenda Laurel writes about meaning as it relates to two of Aristotle's Four Causes. Material causes are inductive, while formal are deductive. "These two causal forces are at work simultaneously, rather like taking inductive and deductive approaches simultaneously in problem-solving."

[2] The variations that occur in interactive storytelling systems are guided by user input. Unlike the interactive story, story generators are not concerned with user agency. The management of variations for interactive stories, on the other hand, should yield meaningful responses to the user. Recall that for story generators, models of storytelling (such as author, world, or reader models) manage the space of variations. For story generators, variations derive their meaning from such models and are less concerned with user-interactions.

(Laurel 2004). This following section identifies a few examples of modeling meaning.

In the Restaurant game, Jeff Orkin collected the actions of human players and learned the conventions of restaurant behavior. Using a select variety of material affordances, he gave the user various avenues to interact with this virtual restaurant. He was then able to use the patterns of user behavior to inform how a computer would simulate higher-order activity. The game creates meaning by assigning goals that require predetermined patterns of behavior (Orkin, 2007).

In his article, Preliminary Poetics, Michael Mateas draws the connection that formal affordances from sophisticated stories would require counter-balance from material affordances (Mateas, 2001). In Final Fantasy, for instance, the player's goal is to accomplish battles and quests that move the story forward, having little influence on the story itself. The Sims, on the other hand, enforces patterns of behavior, giving more variability of outcomes through abstract models and serendipity, heavily dependent on human interpretation.

Mateas argues that the interactive drama Facade realizes a more dramatically compelling interactive experience than found in Final Fantasy and the Sims. Facade provides material affordances that enable exploration through a preauthored history of a married couple (Mateas, 2005).

As their mutual friend and through your virtual interactions, you shape the last 10-20 minutes of their relationship, for better or worse. This creative use of story enables Facade to allow traditional authoring for the backstory, while still permitting a number of material affordances through a social situation that is procedurally communicated to the player. This interactive drama gives an uncanny sense of living out the virtual situation.

Brenda Laurel labels the story of the user's decisions and experience the player's plot: "The authorship of the designer(s) is of a different order than the creative inputs of the player; the designer authors the world and its affordances, while the player creates a distinct path through the game world that can be said to be the player's plot." Seeing the player's plot as an exploration of pre-authored content helps to create helpful design guidelines. As indicated earlier, the player's plot is the story generated by the user-guided variations.

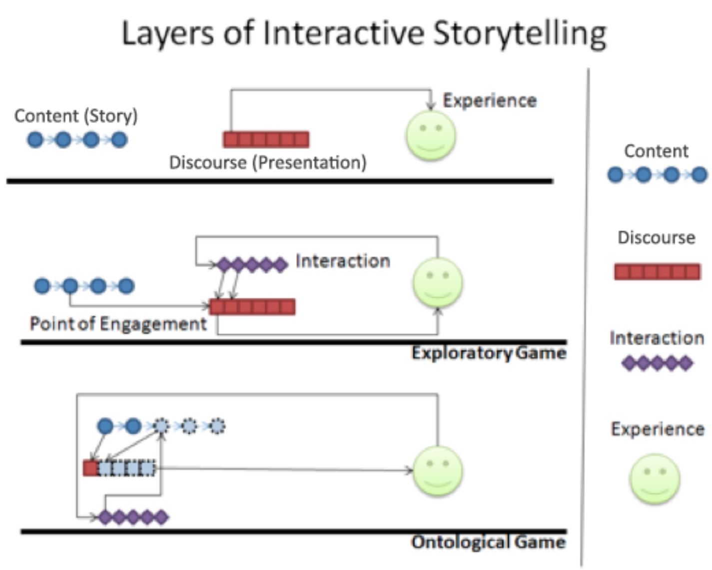

Figure 2-1. Layers of Interactive Storytelling.

Figure 2-1 above shows four aspects of an interactive story: the story, presentation, interaction, and experience. The top diagram indicates a non-interactive experience, the middle, an exploratory game, and the bottom, an ontological game. A non-interactive experience is unaltered, while the ontological experience can influence the story structure. Typically, in exploratory games, the fate of the world has been determined, and the player is able to discover the details from the pre-authored referable content.

Janet Murray writes about the power of exploratory interactions in what she calls multiform storytelling. To represent this structural formalization, areas such as computational cinematography aim to operationalize presentation (or discourse). Work in this area focuses on camera angles of 3-d worlds and procedurally generated script representation. The Curveship system, developed by Nick Montfort, seeks to operationalize discourse for interactive fiction, as defined by narratologist, Gerald Gennette (Montfort, 2009; Gennette, 1979). These projects establish more applied literary formalizations for narrative design, as Cavazza suggested.

Such approaches, rather than focus on generating the story as a whole or on generating plot, are more author-based—they create leverage by appealing to the authorial experience, by allowing to author to have a more linear authorship. Variations, therefore, come from the structural definition of discourse, and quality, instead of being solely generated, is also (if not more) maintained by the predetermined plot.

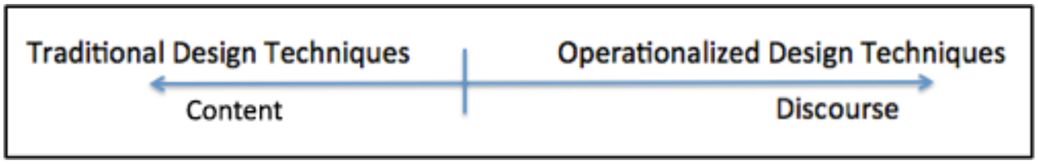

Figure 2-2. Operationalizing the Story/Discourse dichotomy.

2.3 Referable Content – Managing Variations

In this sub-section, we further apply Marie Laure-Ryan's theory on modes of play for insights on game design for generative story systems. Seeing Facade as an exploratory game of preauthored backstory helps in understanding the authorial affordances. Part of Facade's challenge is allowing the player's plot or the exploration to become extensions of the backstory; therefore, referable story content. This blurs the lines between story (content) and presentation (discourse), ontological and exploratory, fabula and syuzhet.

The diagram in the Figure 2-3 below places Myst on the top and Façade on the bottom. According to Ryan's definition, the top is exploratory while the bottom, clearly, is ontological. Our diagram, however, shows that there is less of an ontological/exploratory binary between the two games; rather, there is a relationship between the player's plot and the referable content. On the left, we have referable content, or the fate of the world. On the right, we have the player's plot, which can be seen as the player's fate. In our revision of Ryan's modes of play, it is more helpful to designate all aspects of play as being ontological, since they affect the player's fate; in contrast, accessing referable content is exploratory play. What makes a game ontological or exploratory, then, is how much the player's fate impacts the fate of the world (similar to Ryan's original definition). In the bottom diagram, as the player's plot progresses, the storytelling AI is able to represent a new world each time, adding the events caused by the player to the referable content of the story world. The main idea is that exploratory play (shown on the left side of the diagram) accesses content that can no longer be changed, while ontological play (on the right) allows the experiences-to-come to be altered.

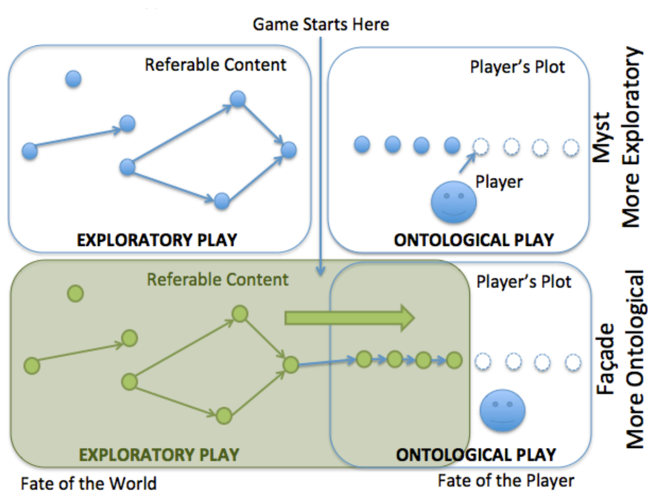

Figure 2-3. Referable content as stories progress.

Recall that we tied exploratory play to discourse variations and ontological play to plot variations. In the beginning of this section, we explained the origins of this content/discourse dichotomy as it relates to generative storytelling. Researchers in this area are looking for ways of improving the user-end experience: for example, having computers tell better stories, or creating games with a greater sense of user-agency. Most storytelling AI uses technology to create sophisticated forms of plot variations (neo-poetics), as this increases user-agency and is often seen as the most obvious way to reduce authorial burden.

Structuralism paved the way to the theoretical separating of plot and presentation. In practice, more recent AI work has been focused on discourse variations. This creates two distinct sandboxes, like the Myst figure suggests. With no overlapping, the box on the left is the fixed plot, and the box on the right represents the opportunities to access this content. This application of literary formalizations creates more straightforward design guidelines for authors. Instead of directing all focus towards the plot, these approaches acknowledge the plot, set it aside, and innovate around it.

From the player's point of view, the content/discourse dichotomy is appropriate; however, as we see in Figure 2-3 above, in sophisticated AI systems, notions of ontological (content variation) and exploratory (discourse variation) overlap. Additional answers that help inform the design process come from asking which plot(s) we are impacting, what level of variation is being managed, and how does the system understand the difference. The content/discourse dichotomy usefully informs game studies, but is limiting in regards to game design. Digital and interactive experiences amplify the complexities of storytelling, showing how designing for universal constructs of content and discourse is a clearer but still nebulous design endeavor.

2.4 Satellites and Kernels – Authoring Variations

One way to parameterize these operations would be through ideologies. We see this often in justice trials, mystery stories, and government propaganda. An early fictional example is from Kurosawa's film Rashomon. This story creates what is called the Rashomon effect, or the unreliable narrator. Systems that model ideology such as Goldwater Machine, Politics, and Terminal Time use rhetorical operations to match belief systems instead of solely generating events and discourse (Abelson, 1963; Carbonell, 1978; Mateas, 1999). In the case of unreliable narrator, the event-discourse dichotomy becomes problematic. For example, if there are rules that allow the narrator to make up events outside of the referable content, then is this considered plot variation or discourse variation? What if the narrator simply hyperbolizes some existing event?

From the design perspective, operationalizing discourse is a straightforward way to lighten the authoring challenges with AI systems. It is, however, even more helpful to consider the operationalizing of supplementary variations, rather than discourse variations. Supplementary variations are non-essential to the central constructs of the world, which we call constituent events. Consequently, the revised Façade diagram now has two non-overlapping constructs: constituent events and supplementary variations. Whereas in the Curveship approach, the plot of the story is acknowledged and then innovated around, here we acknowledge constituent events as author specifications for the story world and innovate around everything else.

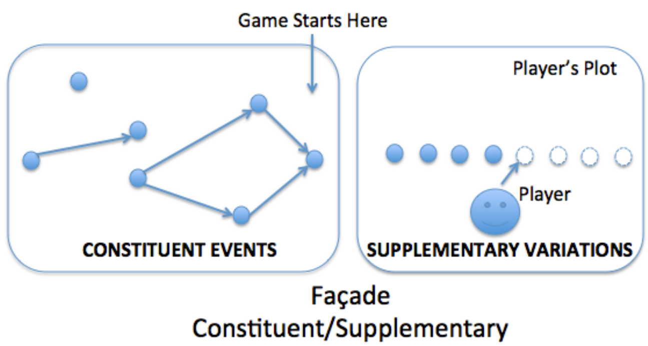

Figure 2-4. Representing Constituent and Supplementary story elements in Façade.

We shift our discussion from plot and discourse to supplementary and constituent. Rather than authoring tiny pieces of plot and operationalizing everything else, this approach aims to author, by convention, portions of the story world called constituent events. Models of storytelling, similar to the story generation tradition, can manage the space of supplementary variations. For interactive story, supplementary variations can be operationalized, similar to how Curveship does for managing discourse variations.

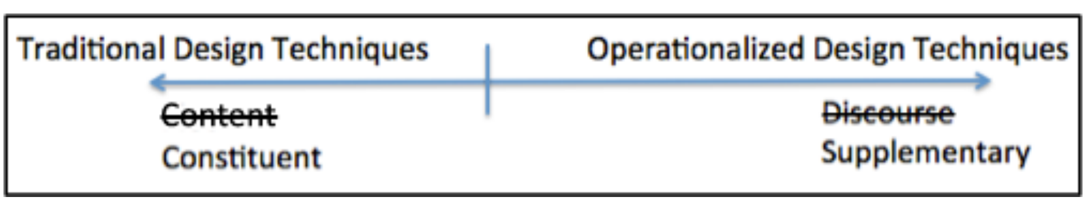

Figure 2-5. Operationalizing the Constituent /Supplementary dichotomy.

In Figure 2-5 above, we replace the plot and discourse dichotomy with the constituent/supplementary dichotomy. Operationalized supplementary variations may require models of character motivation, theories on perception of innocence and guilt, dramatic causality, and relationships. Social norms, such as, "people own things" and "you shouldn't take what doesn't belong to you" may need to be scripted in or abstractly defined.

Given a story, we can imagine a series of expensive animations that portray the constituent events. Through operationalized supplementary variations, we have freedom to designate variation-types, perhaps only requiring low-cost assets or assets that the computer can easily generate.[3] This reduces the storytelling problem to a problem of "how we achieve novel end-experiences in light of constituent constraints." The solutions of this particular problem become a subset of approaches in storytelling AI, one that considers creating the least amount of burden on the author first, and shaping the user experience second.

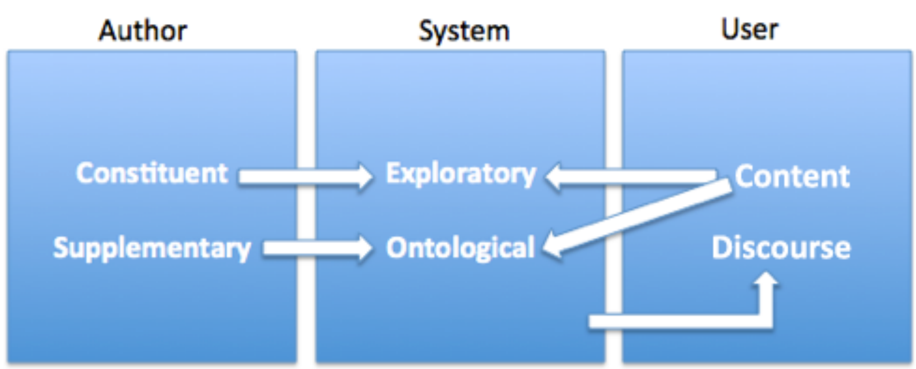

Figure 2-6. Theoretical breakdown from the author, system, and user's point of views.

[3] [footnote on quest generation and Reidl's work in script interpretation]

Finally, in the above, we show how the author and user collaborate around a generative story system. The author's constituent specifications create referable content that the player can explore. Operationalized supplementary variations can shape the experience around the user's understanding, or, if interactive, agency. In regards to agency, the player's plot contains the actions and decisions of the user, which inform which parts of the story are retrieved and how the world adapts—represented by the two white arrows pointing from user's plot. The system then creates feedback for the user, symbolized by the user's presentation or discourse. Conclusively, in comparing this new design model with the content/discourse model, sophisticated systems, like Façade, are more clearly understood and designed.

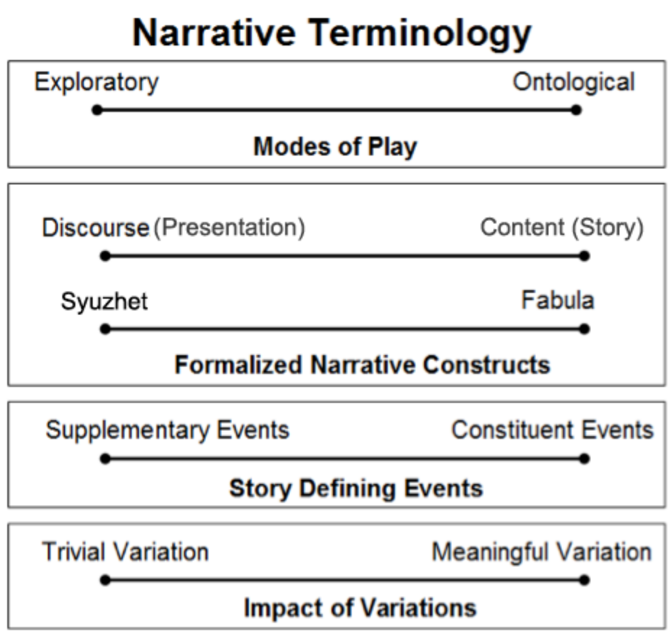

Figure 2-7. An overall distinction of story and discourse.

To provide a better means for analysis, a new distinction should be made. As we have described, Supplementary and Constituent is another way to measure when some storytelling conventions goes from exploratory to significantly impacting the story. The portrayal of constituent events is what builds the skeleton for how the story unfolds, while all else is padding and decoration.

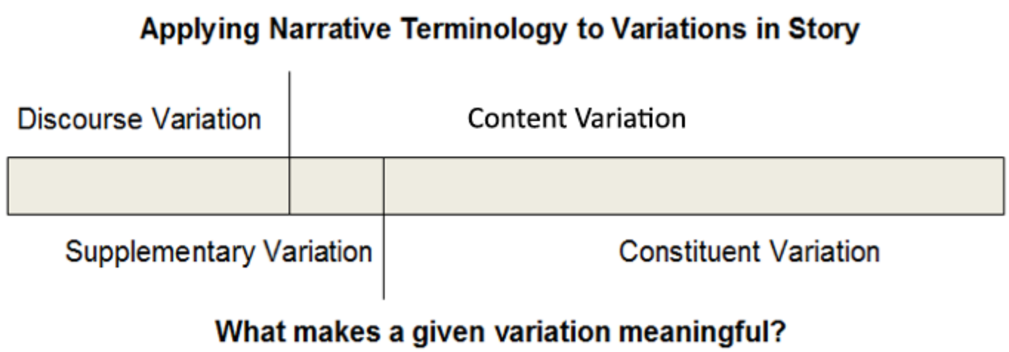

Figure 2-8. Applying narrative terminology to variations.

In this chapter, we not only make the case for narratology driven approaches, we identify the specific dichotomy of satellites and kernels (or constituent and supplementary) as an optimal framing for storytelling AI. Finally, we elevate the authoring of variations rather than primarily looking at content and discourse, which becomes the basis for design and discussing around the contributed systems in this work.

In the next chapters, we analyze the literature on storytelling. We talk about foundations of storytelling and games, and explain further what is meant by interactive and intelligent stories. This dissertation focuses on digital interactive and intelligent narratives, as illustrated in the diagram below. For example, Façade, being a computer based interactive story that makes decisions in game, is interactive, intelligent, and digital, while Shakespearean plays are non-interactive, non-intelligent, and non-digital.

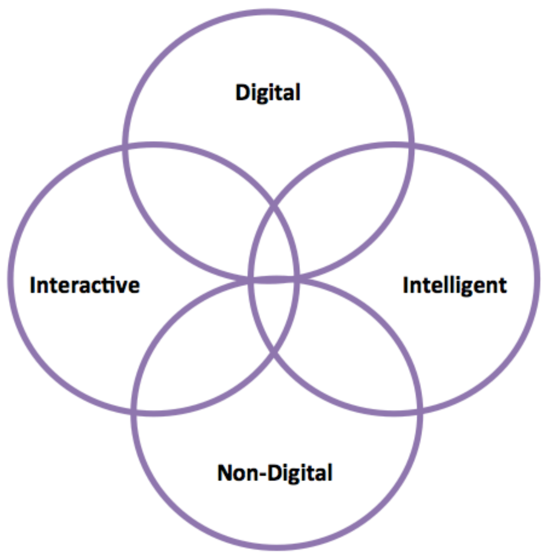

Figure 2-9. Story Experiences: interactive versus intelligent.

Understanding the previous work around linear narrative, interactive narrative, and artificially intelligent narrative, gives additional avenues for evaluating storytelling tools and systems. Table 2-2 below, takes a look at the generalized leverage that each type of story provides for its author.

	Linear Story	**Branching Story**	**Current AI Approaches**	**Ideal AI System**

Quality	High	High	Low-High	High
Variations	Low	High	High	High
Control	High	Low	Low-High	High
Effort	Low	High	High	Low

Table 2-2. An in-general overview of storytelling models.

The final column is a representation of success for the ideal AI story system. Such an ideal AI system would enable an author to produce quality stories, with meaningful variations, through the right controls and system flexibility, without increasing the authorial burden and reducing effort. This dissertation aims to advance the pursuit of storytelling AI as framed by the evaluation metrics outlined in Table 2-2. In order to have a framework for evaluation, the following short chapters discuss a more in depth background for framing the problem.

Chapter 3 – Linear Storytelling Background

To understand improvements in story, we will define what a story is or what we mean by story. The story term takes on various meanings in the literature, so, instead, we use the term narrative. Structurally speaking, a narrative is a set of truths or events and their presentation. As described above, we understand that a narrative is made up of the two parts, (story) content and discourse. The basis for such a framework dates as far back as ancient philosophy and Aristotelian poetics. Theorists have since built upon these formalisms to abstract and define narrative. Conclusively, the content and discourse framework is well suited for the common occurrences of stories found in novels and movies, but for interactive spaces like in games, there needs to be a slightly different approach to understanding narrative.

3.1 Storytelling as Authoring Events

To understand the difference between storytelling from narrative in contrast to storytelling from interactive narrative, we first look at the former. In his book The Theory and Analysis of Drama, Manfred Pfister references Aristotle's poetics, the earliest surviving work of dramatic theory, stating that 'story' should be formally defined as having three required ingredients: "one or more human or anthropomorphic subjects, a temporal dimension indicating the passing of time, and a spatial dimension giving a sense of space." He then separates the subject from its presentation by citing Russian Formalist

Tomashevski's fabula and syuzhet concepts, which he calls "story" and "plot." He further breaks down stories into actions, action sequences, and events (Pfister, 1977). Seymour Chatman, writing on narrative structure in fiction and film, furthers the concepts of fabula and syuzhet, or the "raw materials of a story" and the "organization of a story" in his book Story and Discourse.

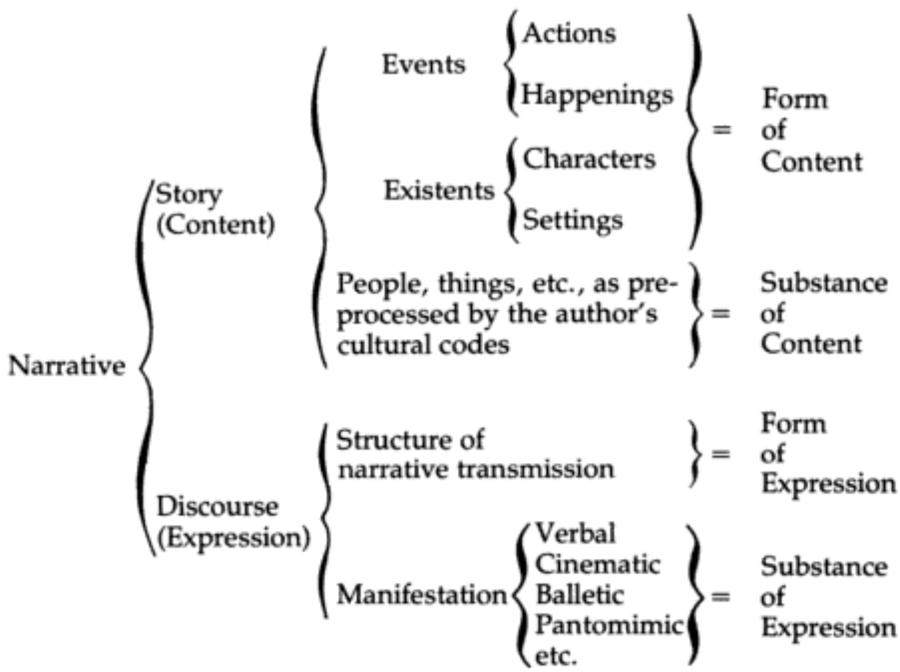

Figure 3-1. Formal breakdown of narrative. (Chatman, 1999)

As illustrated in the diagram above, Chatman formalizes story as events and existents. Rather than simply citing a set of actions or happenings, Chatman includes properties of sequence, contingency, and causality for story, while keeping these properties distinct from the discourse (Chatman, 1978). Gerard Gennette, in his book Narrative Discourse, identifies specific components of discourse: order, duration, frequency, mood, and voice. All aspects of discourse are possible variations on the presentation of a story (Gennette, 1979). In summarizing the previous work, we settle on the framework below for understanding Narratives.

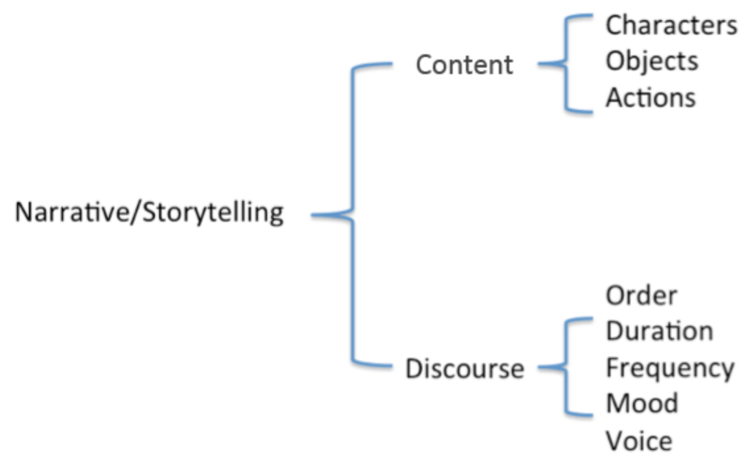

Figure 3-2. The framework for narrative that we will be using for this work

From movies to theater to film, there is a clear distinction between story content and discourse. Narratologist H. Porter Abbott concludes that "narrative is the representation of an event or a series of events," the series of events being the story and the representation being the discourse (Abbott, 2002). Given this definition, how does narrative carry over into the space of interactive experiences? So far, we've laid out the basic building blocks for storytelling, and although this framework also applies to the stories we get from games, the author/audience relationship is a bit more complicated when stories become interactive. Even in linear narratives, likely found in books and movies, there's notable dynamics between the telling and receiving of story. We explain in the next section how audience perspective is driven by events, and the author's perspective is driven by variations.

3.2 Storytelling as Authoring Variations

Many researchers and developers have used computing to enhance storytelling, with varying degrees of success (Cavazza, 1999). When assessing the outcomes of storytelling practices, it is common to evaluate the end result from the audience's view, a very event-focused perspective. In contrast, we want to view storytelling from the authorial view, which is a variations-based perspective.

First, where do we find variations in story? In games like Mass Effect or Planescape Torment, when the user is required to give a verbal response to another character, the game should respond appropriately to the selected statement. When one decision changes any aspect of the experience, a variation must have been designed into the experience. Decisions can also be on the game side, such as the probabilistically distributed enemies that you find in a dungeon for a Final Fantasy game or the drama-managed response from Grace to Trip in Façade.

Narratologists and psychologists have realized, however, that variation occurs even in the absence of intentionally interactive experiences. Variation will always exist for at least two points of a storytelling, the delivery and the reception—the delivery of the author/teller and the reception by the audience.

Professor and screenwriter, Robert McKee, points out that storytelling is as much about how a story is decidedly told as it is about the story itself, "A story is not only what you have to say but how you say it." He describes the art form as being about principles of eternal and universal forms, archetypes, realities, and mastering the art with originality and respect for the audience (McKee, 1997).

Authoring is driven by choice; therefore, the author's experience in storytelling would be appropriately represented through variations. The reception of a narrative is driven by outcomes, and is easily represented by series of events. Still, both author and audience experience variations in storytelling. The author chooses how a story is told, while the audience chooses how they will understand it. Narratologist, H. Porter Abbott, asserts that variations are unavoidable byproducts of the storytelling exchange. In particular, the occurrence of occlusions is inevitable in storytelling. To be able to exhaust the details of a story is not only unnecessary, it is impossible. As long as the constituent events are communicated, the lack or abundance of supplementary details is up to the storyteller.

> Narratives, by their nature, are riddled with gaps. Even if we come as close as we humanly can to avoid underreading and overreading, we still have to fill things in if we are to make sense of the narratives we read or see (Abbott, 2002).

The way we use our presentation of the story content or use discourse to conveys the events is not entirely without meaning, although in the past, it has been seen as secondary to the story events. Similarly, supplementary events can have profound impact on how a story is communicated. Rhetoric, for instance, is an unavoidable byproduct of narrative, and by default is used for normalizing a sequence of events.

> The impression of causation that we have been examining is one of the ways - a powerful one - of suggesting normality. But we can extend the rhetorical leverage of normalizing to many other features of narrativity. In that sense, narrative could be called a kind of "rhetoric of the real" in that it accounts for things. You could in fact argue, and people have, that our need for narrative form is so strong that we don't really believe something is true unless we can see it as a story. Bringing a collection or naturalizing narrative coherence can be described as a way for normalizing or naturalizing those events. It renders them plausible, allowing one to see how they all "belong" (Abbott, 2002).

Psychologist Frederick Bartlett similarly investigated this idea of dramatic points of view, through understanding cultural contexts of social groups.

> It often happens that a folk-story which has been developed in a certain social group gets passed on to another which possesses different habits of life and thought, different social institutions, customs, beliefs, and belongs to a widely divergent level of development.
>
> Thereupon A, repeating the story to B, involuntarily introduces slight changes, perhaps replacing the name of an object which, he has rarely or never seen by that of some other object with which he is familiar.
>
> B carries on the same process, and in this manner, by means of a number of alterations, many of them apparently trivial in nature, the material is gradually reduced to a relatively fixed form which, congenial to its new environment, bears only what may be called a "family likeness" to the story as found in the other community (Bartlett, 1932).

Not only does bias, or variation, exist in the telling of story as Schank describes in "story-fitting," it exists in the audience understanding of story, or, as Abbott calls it, overreading and underreading. Finally, these biases create substantial variation, most likely a heavy by-product of our point of view and context of being, as Bartlett demonstrates. A proactive author takes into consideration the space of interpretations, and manages these variations through how they tell their story.

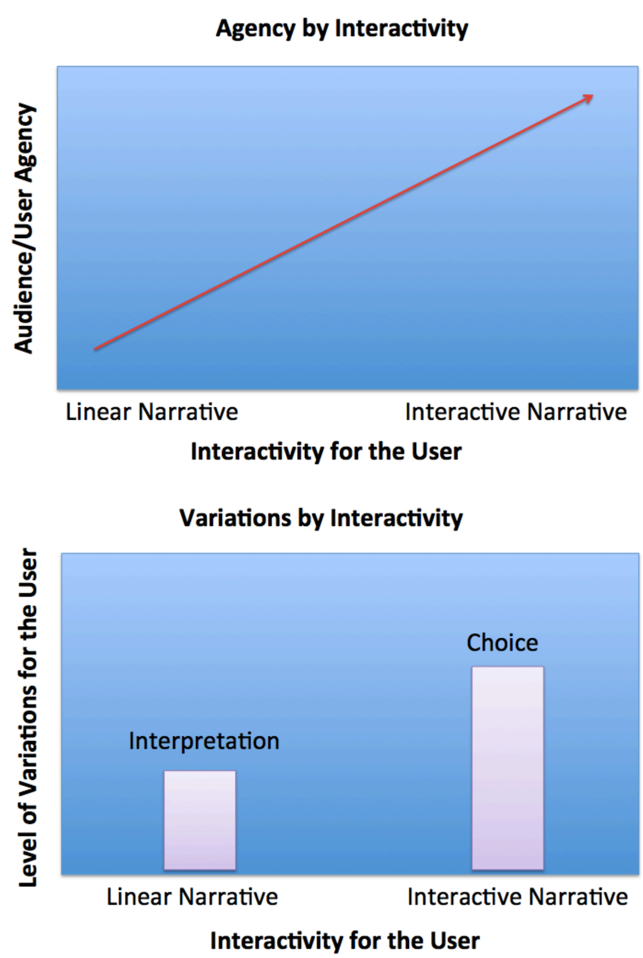

Figure 3-3. Variations exist in both the interpretations and the choices of the audience

In the case of linear stories, the audience only has agency in a story's interpretation. Even if the author were constrained to only relay honest recounts of actual events, it's still as much (if not more) agency than merely interpreting what is conveyed. It's when the audience is given the ability to make choices for the narrative that the author gives over some of their agency in dictating outcomes to the audience. Agency becomes more of a significant factor as the storytelling interaction becomes more complexly interactive. As we add functionality and tools to our stories, the author's responsibilities change quite a bit. In the diagrams above, we show that agency and variations are limited in linear experiences in comparison to interactive counterparts. To author meaningful interactive choices for the narrative, we discuss the ontological/exploratory dichotomy for Interactive Narratives in the next chapter.

Chapter 4 – Interactive Storytelling Background

By giving the audience a choice, storytelling becomes more than just events. The audience takes part in authoring their own experience steered by the switches and knobs designed by the author. While the audience still experiences story events, the storytelling aspects for games can be broken down into two modes of play, derived from the content/discourse dichotomy. Now, the author must also be mindful of the variation space that the audience faces, in addition to the spaces of possibilities inherent to the authoring itself. As interactions become more meaningful, there's even more of an obvious shift from storytelling as the management of events to the management of variations.

4.1 Interactive Storytelling as Modes of Play

How have questions about variations in story and user experience traditionally been addressed in the literature? Marie-Laure Ryan provides a model of ontological and exploratory modes of play to distinguish between the types of variation that can occur in story, specifically for games. In the exploratory mode, the user is free to move around the database of story content, but this activity does not make history or alter the story; the user has no impact on the destiny of the virtual world. In the ontological mode, by contrast, the user's decisions send the history of the virtual world on different forking paths—they determine which possible world, and consequently which story, will develop from the situation in which the choice presents itself. According to Ryan, an interaction can either be exploratory, where the story is unaffected, or ontological, where the user can alter the destiny of the world.

Interactive media gives the audience the ability to direct their experience in a story world. To a player, exploratory modes of interaction, which produce variations that do not change the fate of the world, can be less meaningful than ontological interactions, which produce variations that can alter the story. According to Ryan, the puzzle-solving computer game Myst is an example of exploratory play, while the Choose Your Own Adventure book is an example of ontological play. Given more agency to the audience, interactive stories and games create spaces for more dramatically compelling interactions or “modes of play.” Therefore, as mentioned in Chapter 1, creators aim for a greater ontological mode of interaction, the Holy Grail for interactive storytelling (Ryan, 2001). Table 4-1 below shows Ryan’s examples of ontological and exploratory.

Ontological Play	The best known example of a narrative system with an ontological/external type of interactivity is the series of children books Choose Your Own Adventure. The underlying structure of these stories is a tree-shaped diagram, on which each branch is kept separate from the others. This enables the designer to maintain a strict control over the linear sequence of events.
Exploratory Play	The mystery story, in which two narrative levels are connected: one constituted by the actions of the detective, the other by the story to be reconstructed. In this case, one level is predetermined, while the other is created in real time by the actions of the user. Example: the computer game Myst, where the user explores an island and solves certain puzzles in order to crack the mystery of what happened in the past.

Table 4-1. Modes of Play

In practice, most story generators and drama managers aim to construct compelling ontological variations of story, subsuming its exploratory or presentation aspects.[4] Recently, however, Nick Montfort designed a system called Curveship that isolates the presentation aspects of telling a story (Montfort, 2009). Montfort draws a line for where variation in experience is on the discourse level or not. He identifies a set of interactions, modeled as

[4] Joseph Bates defined the process of presenting plot points to a user as refinement, a nontrivial challenge in designing intelligently interactive experiences (Weyhrauch & Bates, 1997).

discourse by narratologist Gerard Genette; therefore, any variation on the discourse level is not a variation of story, or "exploratory."

> The study of narrating, of how the same underlying events can be told in different ways, has been undertaken systematically in the field of narrative theory or narratology, in which the distinction between story/content and discourse, between that which is narrated the narrative itself, has been central. (Montfort, 2009)

Montfort accepts the traditional story/discourse distinction (according to which the act of narrating takes a story and turns it into discourse). In his system, the computer performs the act of narrating. Specifically, Montfort leverages variations that are introduced through temporal reorderings from the traditional narratology literature. With Curveship, he has produced a system that can take the same story (sequence of events) and narrate them differently.

Other than in systems like Curveship, where there the space of variations are intended to be solely exploratory, authoring stories is not so clearly cut into ontological and exploratory modes of play— where it is either in one mode, allowing your audience to alter the fate of the world, or only permit its exploration. At different points within the same story, there could be events that are ontological and events that are exploratory. Ontological events cause the world to veer off into other directions (high agency), while exploratory events are meant to only give information (low agency). In contrast to linear stories, the audience is now given choice, and can experience the world through modes of play, having to manage and enact events of differing degrees of consequence.

Likewise, the author is no longer building one story, but the opportunity for many stories within the space of possibilities or permissible degrees of agency.

While discourse events are easily identified and capture a specific type of interaction, ontological events cover a broader space of possibilities. When we consider an edge case, such as foldback (which we discuss in the next section), ontological variation is permitted, yet the outcomes are non-significant, since all paths converge. Applying the definition that Marie-Laure Ryan gives, there is an unaccounted case, where the user makes history by changing the story and yet does not change the fate of the virtual world.

4.2 Interactive Storytelling as Meaningful Variations

Game researcher Chris Crawford examines typical story models that are found in game-based interactive stories (Crawford, 2012).

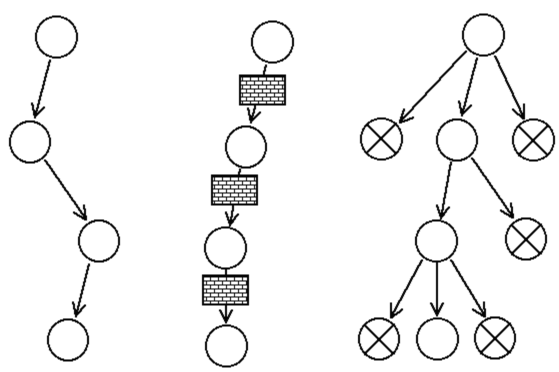

Figure 4-1. Methods of Interactive Storytelling (Crawford)

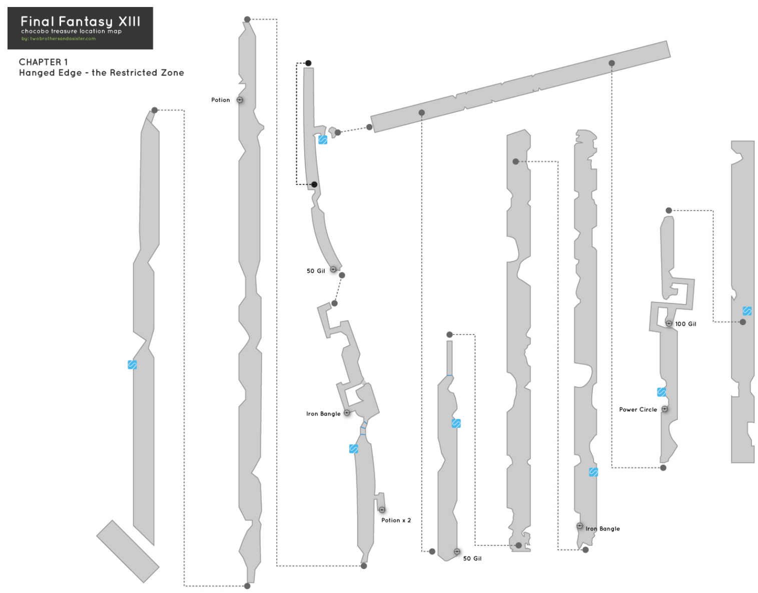

Figure 4-2. Final Fantasy 13's linear structure.

The image on the left of Figure 4-1 above depicts a linear storyline. Many story games have linear story structures, with the interactivity involving non-story-based activities such as combat. Games that support story variations often make use of the second puzzle-solving model in the middle or the all paths are dead-ends except for the correct one model depicted on the right. Chris calls these two models "Obstructionist Stories" and "Kill 'em if They Stray Stories."

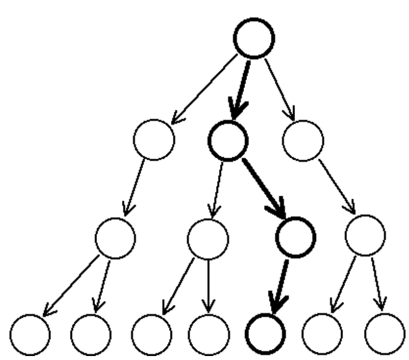

Figure 4-3. Branching bottleneck.

Figure 4-3 depicts the typical approach for modeling the causality of interactive stories with ontological variation using story trees. This is how CYOA books are authored. With enough effort, deep and compelling experiences can intelligently respond to user actions. The issue with this approach is that it only works for fairly shallow experiences. Chris makes this observation about storytrees and their authorial burden.

> Let's be conservative, then, and assume that our interactive story needs only a hundred events or actions in it. In other words, there will be one hundred plies in our interactive story. Let us further assume that each branchpoint will have only two choices available to it this is the absolute minimum required. This implies that the storytree will have a total of 2^100 nodes in it. How many is that? About 10^30. If you had a billion employees creating nodes, each one making one node every second, working 24 hours per day, 365 days per year, then it would take 30 trillion years to make the nodes necessary to build that one storytree (Crawford, 2012).

Given the exponential authoring explosion of the storytree approach, another typical model employed in storygames is foldback, as shown in Figure 4-4 below. Instead of blocking the choices or not giving choices, choices can make minor changes in the storyworld then branch back to an effectively linear storyline.

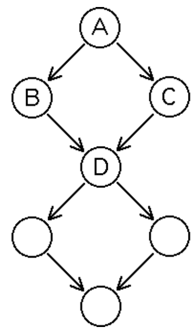

Figure 4-4. Foldback

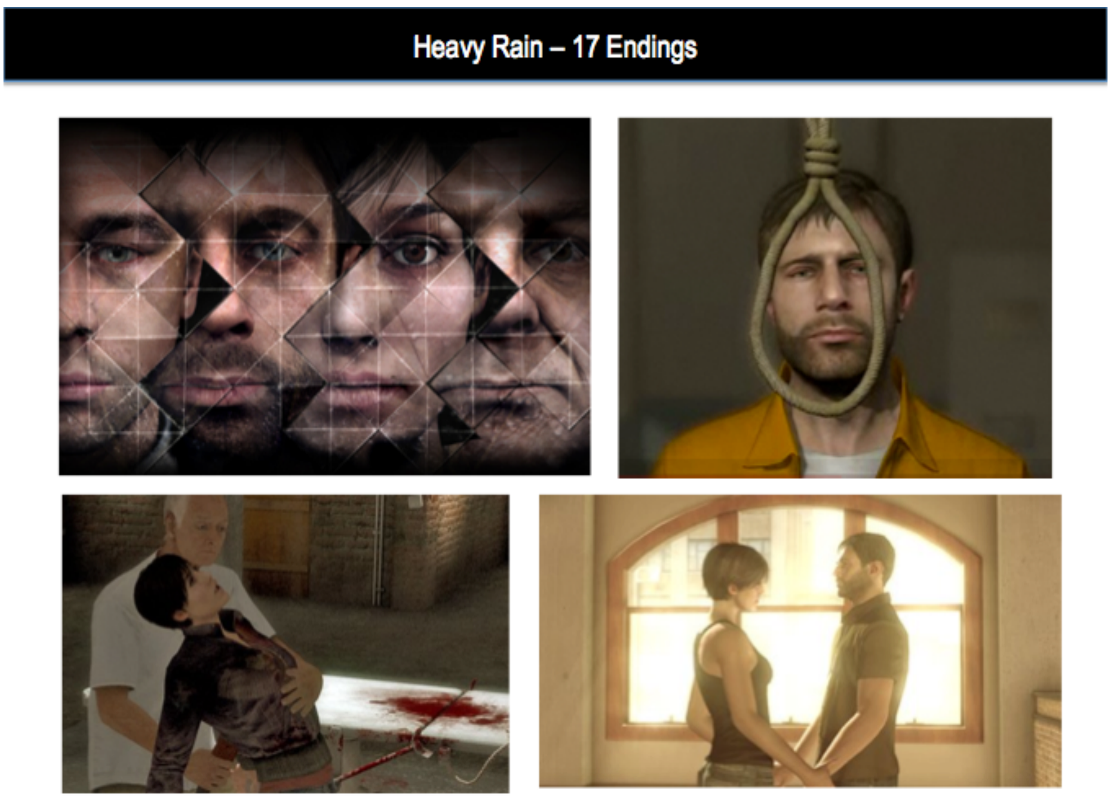

Figure 4-5. Different outcomes for the conclusion of Heavy Rain.

For example, take the game Heavy Rain. This game has 17 endings, however, there is little branching in the story until the end of the game. This approach makes use of foldback through most of the story experience, only introducing true branching at the end, as depicted in Figure 4-6 below. In contrast to the multiple endings, there are numerous supplementary interactions like what is now known as "press X to Jason." At the beginning of the game, as the father, the player can walk around a shopping mall, calling out to his son, "Jason," with every X-button press. This yields no alternate consequences and foldsback to the main story no matter how the player enacts with the scene.

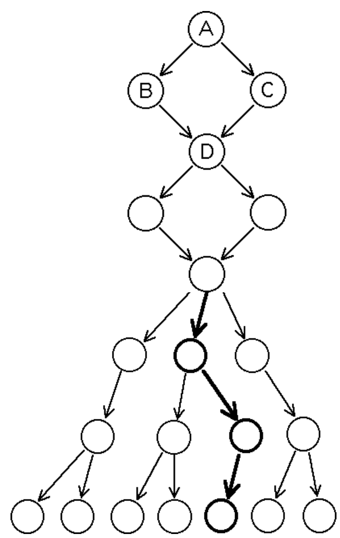

Figure 4-6. Reducing branching factor, by starting with foldback and introducing branching at the very end.

In the distinction between ontological and exploratory variation, foldback is an edge case, altering the story without changing the fate of the world. Crawford describes foldback as "fraudulent" interactivity, since this model is merely masquerading as a story-tree. Rather than using the ontological/exploratory distinction, foldback is more easily understood using Abott's constituent/supplemental distinction described (Abbott, 2002). This is an interesting edge-case in interactive story design to be addressed in the next chapters as we discuss types of variation in story.

The nature of interactive stories is that they provide a variety of experiences dictated by some amount of feedback from its audience. A big challenge is the authoring process that goes into creating meaningful variations. As discussed, the authoring process of creating non-"fraudulent" interaction creates an authoring burden that increases exponentially. In current research, the application of Artificial Intelligence (AI) towards storytelling aims to alleviate the burden and to create better story experiences for the user whether through drama management, story generation, or authorial tools. Such systems include: Declarative Optimization-Based Drama Management (DODM) (Sullivan et al., 2009), IPOCL (Riedl & Young, 2004), Riu (Ontanon & Zhu, 2010), Mexica (y Perez & Sharples, 2004), Minstrel (Turner, 1993), StoryCanvas (Skorupski & Mateas, 2010), and many others.

The next chapter gives a brief definition for what we mean by AI, diving specifically into the previous work that is most useful towards understanding the contributions of this dissertation.

Chapter 5 – Storytelling AI Background

As we have seen, non-intelligent stories have several limitations, including a lack of agency (for linear stories), an exponential authorial burden (for branching stories), and rigid or inconsequential interactions (like foldback in interactive stories). We now move to a discussion of the background for intelligent stories and systems, which are stories that incorporate Artificial Intelligence towards the authoring or adaptation for their audience, and tools that enable authors to shape their desired narrative system or world. Such narratives are artificially constructed, rather than naturally composed entirely by a human author. Understanding the synthetic aspects of these narratives would not be complete without a discussion of believability. As with any generative, expressive, and creative AI, we understand the performance up to and beyond the uncanny valley through the lens of believability.

Since the coining of the term, there have been many pursuits towards AI. We are interested in previous work that supports the approach we are taking towards AI for storytelling. In this chapter, we review what's been discussed so far to build towards the technological motivation of believability in AI, then illustrate the previous work done, starting with early chatbots, then Goldwater Machine, and finally Terminal Time.

5.1 Intelligent Narratives in terms of what we've discussed so far

A story can be interactive, and a story can be intelligent. These two characteristics are related, but not synonymous. Interactive does not imply intelligent, and vice versa. As Figure 5-1 below shows, interactivity is also not constrained by digital or non-digital mediums. Some examples of non-digital and interactive stories would be Choose Your Own Adventure books, Live Action Role Playing games, Dinner Mystery Theater, and Table-Top RPGs. Video games, like Super Mario Bros and Final Fantasy, are clearly interactive, because they rely on decisions made by a human player. Interactive stories, therefore, take into account human interactions to construct the experience.

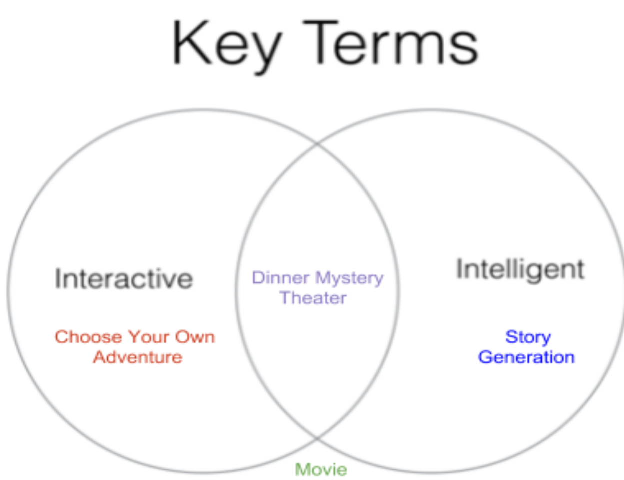

Figure 5-1. Indirect relationship between interactive and intelligent stories.

A Choose Your Own Adventure novel is interactive but not intelligent. The decision points are scripted into the text by the author. In intelligent stories, the experience itself must demonstrate its own ability to make decisions or reason about the story world, whether with a human dungeon master (in a table top

RPG) or a computer algorithm. While it's easier to determine whether an experience is interactive, the standards of human storytelling can help identify whether these experiences are artificially intelligent.

Where are the decisions made when creating a story experience? The computer can generate a storyline, assisting or even taking the place of writing the sequence of events. The computer can generate the game mechanics or the winning conditions (Smith, 2010) that trigger the story. The computer can manage the delivery of the story (Weyhrauch & Bates, 1997). When applying AI, the role of the computer is can facilitate or enhance the desire of a human author at multiple points of engagement with multiple varieties of decision types.

Human Role	**Computer Role**
Writing a story	Generating a story
Designing a game	Generating game mechanics
Dungeon Master, Improv Actor	Managing Experience

Table 5-1. Computers taking on human aspects of storytelling.

So far, we've established an understanding of story experiences through discussing interactive, intelligent, and artificially intelligent stories. In the next section, we will look at the basic building blocks for story and narrative in order to understand what it means to make better stories. For comparison, how do we see improvements in computer graphics and animation? Similarly, how do we know if stories have improved, whether they even need improving, and how they should improve? Below are two images showing the visual improvements in digital storytelling; the first compares Final Fantasy, released in 1987, with

Final Fantasy XIII, released 2009, and the second shows the twenty-year evolution of Pokemon.

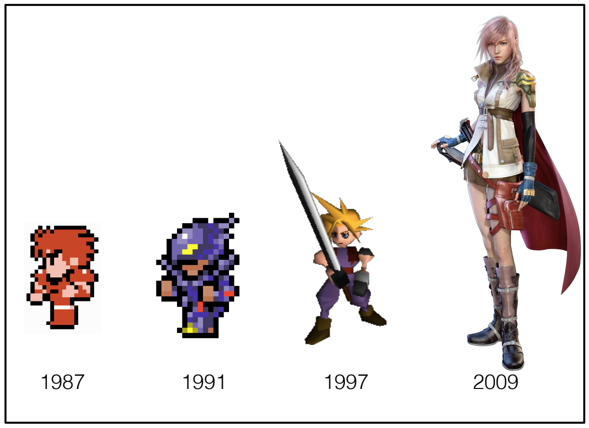

Figure 5-2. Final Fantasy 1, 2, 7, and 13 evolution of character graphics.

Figure 5-3. Evolution of Pikachu throughout the Pokemon series.

http://pokemonspriteguy.blogspot.com/2010/10/25-pikachu.html

In this sub-chapter, we propose a direction for designing and understanding intelligence in stories. Unlike computer graphics, where improvements are visually observable, storytelling can be deconstructed various ways as we've discussed in the previous subsections. Table 5-2 below summarizes the topics covered so far. Chapter 2.1 and 2.3 were about Linear and Interactive Narratives, each having a non-intelligent and intelligent subsection. In Chapter 2.5, we will focus on intuitive advancements for Intelligent Storytelling, which shifts understanding story from the Audience's experience towards understanding storytelling from the Author's experience.

	Non-Intelligent	**Intelligent**
Building Linear Narrative	Authoring Events (Ch 3)	Authoring Variations
Building Interactive Narrative	Managing Modes of Play (Ch 4)	Managing Meaningful Variations

Table 5-2. Representations of chapters 3 and 4 as it relates to Intelligent Narratives (chapter 5)

When building with Artificial Intelligence we are building the Author's experience as much as we are for the Audience. Rather than looking at storytelling as Content and Discourse, or Ontological and Exploratory, we use Constituent and Supplementary as more suitable building blocks. To understand what we mean, we will discuss relevant historical approaches for AI in storytelling.

5.2 Early AI and Believability

Playwright, Robert McKee described the art of storytelling as, "the life story of each and every character offers encyclopedic possibilities; the mark of a master is to select only a few moments but give us a lifetime." The earliest AI attempts for capturing the relevant moments of a character was through chatbots which began with a famous paper in 1950, where Alan Turing asked, “Can machines think?” (Turing, 1950) Turing used human assessment as a means to discern intelligence. Early experiments in believable AI, in the form of chatbots, developed from Turing’s work. It follows that one of AI’s first pursuits was believability. Decades later, on the contrary, the most widely used AI textbook positioned that rationality, independent of human behavior, as the more advantageous pursuit:

> The study of AI as rational agent design therefore has two advantages. First, it is more general than the “laws of thought” approach, because correct inference is only a useful mechanism for achieving rationality, and not a necessary one. Second, it is more amenable to scientific development than approaches based on human behavior or human thought, because the standard of rationality is clearly defined and completely general. Human behavior, on the other hand, is well-adapted for one specific environment and is the product, in part, of a complicated and largely unknown evolutionary process that still may be far from achieving perfection (Russell and Norvig, 1995).

This, of course, excluded storytelling as an advantageous rational pursuit, rather a mere byproduct of human behavior.

Technology, however, exists to extend and augment human capabilities even if we can’t rationally disclose how our capabilities, whether complicated, unknown, or imperfect, function. In support of technology built towards human thought and behavior, storytelling is among the most universal and powerful senses and abilities we have. And if we cannot derive storytelling from base principles, we can at least evaluate the state of the art by comparing the behavior of man to the believability of machine.

5.3 Chatbots: Modeling Reactive Believability

Not only do critics question whether believability is a worthy pursuit, many have argued that Turing's test gives no indication of intelligence in machines[5]. However, at its face value, Turing's Imitation Game quite literally measures whether a machine behaves as a human would. Similar to what McKee describes for a master storyteller, if a machine can masterfully select a few moments that disclose a coherent and novel lifetime, how it is accomplished (intelligent or not) does not change it's authorial goals. Early pursuits developed chatbots as an application intelligence towards human thought and behavior. The following subsections will look at an evolution of AI in storytelling, particularly focusing on character perspectives. We will describe a few systems ranging from chatbots, to political propaganda generation, to historical documentary generation.

Here is a transcript from a conversation with the intelligent and interactive system, Eliza, one of the first and most well known chatbots ever developed.

[5] https://en.wikipedia.org/wiki/Chinese_room

```
Hello. My name is Eliza.  How may I help you?
* I'm fine
> Did you come to me because you are fine?
* No
> Why not?
* I'm not fine
> Do you believe it is normal to be not fine?
* no
> You are being a bit negative.
* no
> Are you saying no just to be negative?
* What is your name?
> Have you asked anyone else?
* what is 2 + 2?
> Are such questions on your mind often?
* goodbye
> OK... "what is 2 + 2". Tell me more.
```

Figure 5-4. Conversation with chatbot, Eliza.

This interactive system could easily be modified into a non-interactive generative story experience. Imagine running two instances of Eliza and having the experience of watching one chat bot chat with another. Even better is creating a counterpart to Eliza. The Atlantic, published a transcript of Eliza talking to Parry the Paranoid agent, another chatbot program[6] (Atlantic, 2014). This is a generative story about a psychotherapist and her patient.

[6] "In 1966, the computer scientist Joseph Weizenbaum created a program that seemed to be a contender for Turing Test passage. The early chatbot searched for keywords in conversations conducted with human typers; if the human used one of those words, the program would use it in its reply. If not, it would offer a generic response. It was meant to mimic a psychotherapist (which allowed it to adopt "the pose of knowing almost nothing of the real world"); he named it ELIZA. And in 1972, the Stanford scientist Kenneth Colby created another program, PARRY—a bot that tried to model the behavior of a paranoid schizophrenic. PARRY was, Colby said, "ELIZA with attitude."" (The Atlantic, 2014)

ELIZA: Good Evening. Tell me your problems.
ELIZA: Please terminate input with a period or a question mark.
PARRY: People get on my nerves sometimes.
ELIZA: I am not sure I understand you fully.
PARRY: You should pay more attention.
ELIZA: Suppose you should pay more attention.
PARRY: You're entitled to your own opinion.
ELIZA: What makes you think I am entitled to my own opinion?
PARRY: It bothers me just to be around people in general.
ELIZA: Please go on.
PARRY: I went to the track at Bay Meadows a while back.
ELIZA: What does that suggest to you?

Figure 5-5. Conversation between two chatbots, ELIZA and PARRY.

From earliest examples of Eliza and Parry, how much have we advanced in the past 50 years? The following two systems we describe all follow inline with this author-centric direction in designing storytelling AI systems. The technical contributions of this dissertation are inspired by the systems described in this literature review and previous work listed in Table 5-3 below. In Chapter 7, we will look at author-centric evaluation metrics towards a framework for goals and measures of success.

System Name	**What it is**
Eliza (1966)	Chatbot
Parry (1972)	Chatbot
Goldwater Machine (1978)	Propaganda Generator
Terminal Time (1999)	Interactive Documentary Generator
Contribution: EMPath (2007)	Drama Managed Adventure Game – **Chapter 6**
Contribution: RoleModel (2012)	Story Generator **– Chapter 8**

Table 5-3. Example list of Intelligent Systems.

5.4 Goldwater Machine: Modeling Systems of Belief.

The birth of the term Artificial Intelligence (AI) happened at Dartmouth College in 1956. In 1963, a very optimistically titled book, "Computer Simulations of Personality," was compiled of a collection of academic papers by psychology theorists. Among those scholars was Robert P. Abelson, whose work has had foundational impact to both AI and Cognitive Science as well as Social Psychology and Political Science.

Abelson theorized that human reasoning was influenced by an additional dimension of factors, directed by our emotions. He named his theory "Hot Cognition," in contrast to Cold Cognition, where our processing of information is independent of our feelings. You could say that Hot Cognition may be less objective, less factual, and less rational in comparison.

Eliza and Parry were meant to deliver an experience of talking to another human being. In the 1960s, early AI scholars proposed systems to model human emotions and affect, defining a future for believable AI. Having a conversation, for example, is not quite the same as telling a story, but in order to model appropriate conversational behavior, the AI must have a basic understanding of human experience. Abelson developed a model specifically for human beliefs and rationalization in a system called the Goldwater Machine.

Goldwater Machine was a system designed and implemented in the 60s. In this early attempt to model human belief systems and rhetoric, Abelson proposed the following definitions:

- A **belief** is "a sentence recoverably stored within an element."
- A **belief system** is "a set of belief-calling elements which are themselves interrelated in a set of sentences."
- A **belief dilemma** is "a situation in which the individual is confronted with the apparent necessity of changing one or more beliefs."

In regards to belief systems, Abelson describes rationalization as the challenge of reconciling an event with one's belief system. He asks, "Is it possible to specify a realistic model for attitude change and resistance to change in sufficient process detail so that a computer could simulate it?" Rather than a general theory of cognition, he is looking for a relational representation between cognition and affect measured by the problem of, "what am I to believe now?" This, he calls attitudinal problem-solving, or how a system ought to respond when simulating an individual who is confronted with a challenge to his belief system. Abelson concludes, "Within the context of attitudes and attitude changes, one might hope to develop a simulation model which would do for hot cognition what others have done for cold cognition." (Abelson, 1963)

Abelson proposes a number of mechanisms to model attitudinal responses or reactions in his paper on Hot Cognition. These mechanisms creatively reconcile gaps and contradictions in a story or belief system:

- **Stopping thinking**, removing particular sentences from thought.
- **Denial**, denying the truth value of the sentence.
- **Rationalization**, the acceptance of the truth value of the sentence, but somehow deflecting its evaluative implications.
- **Differentiation**, the creation of two elements to replace one element.
- **Transcendence**, a difficult higher-order mechanism.
- **Bolstering**, a side process of evaluative change which has the function of compensating for some of the damage done by the imbalance.

- **Evaluative change by default of resistance**, as a consequence of the failure of all other mechanisms, substantial evaluative transfer takes place between the two elements of the key sentence.

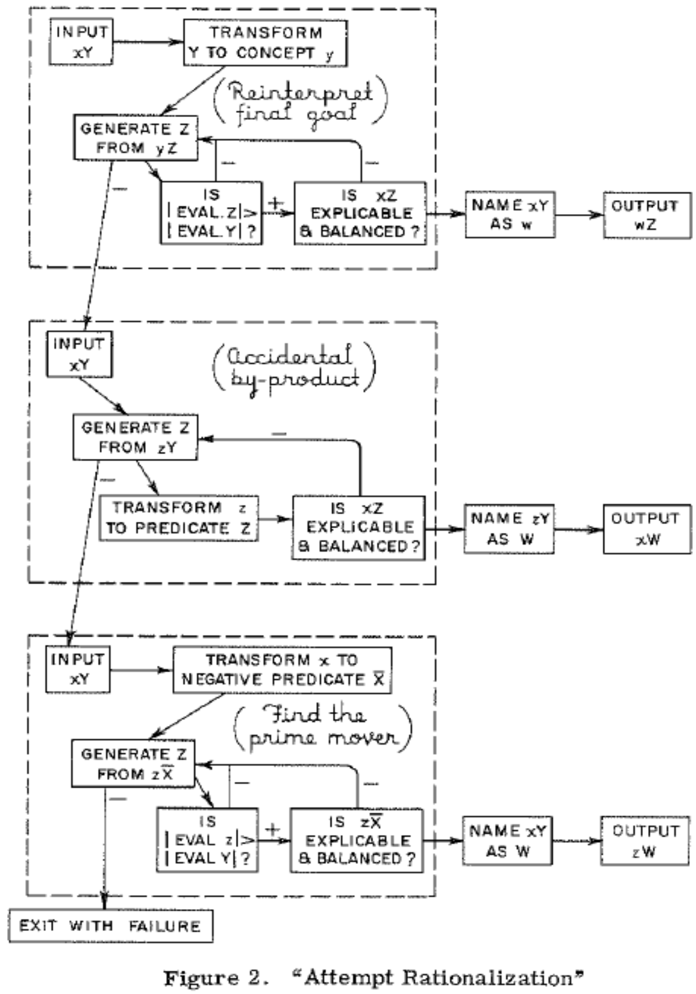

Figure 5-6. Abelson's Rationalization Mechanisms from his paper on Hot Cognition.

Figure 5-6 shows a visual representation of Abelson's three rationalization mechanisms: reinterpret the final goal, establish an accidental by-product, and find the prime mover. In this section, we gave an example of what it looks like to model an AI after human intelligence, with regards to belief systems and rationalizations. These theories became the basis of what was known as the Goldwater Machine. Noah Wardrip-Fruin revisits this work in his book, Expressive Processing, describing the concept of Goldwater from it's inspiration:

> The world seemed polarized to many and, within the United States, names like those of Adlai Stevenson and Barry Goldwater did not simply indicate prominent politicians with occasionally differing philosophies. Goldwater, the Republican nominee for president of the United States in 1964, was an emblematic believer in the idea that the world's polarization was an inevitable result of a struggle between good and evil. (Wardrip-Fruin, 2009)

Towards simulating Hot Cognition, Abelson and his colleague, J. Douglass Carroll, continued to work on the powerful and steadfast aspects of our minds, namely, ideology, rationalization, and bias. The diagram above is a model of human rationalization, drawn by Abelson in 1963, to caricaturize our desire to be "right" or "good." Given a situation that contradicts our belief system, we are confronted with "the apparent necessity of changing one or more beliefs." Our resistance to this change can be formalized as "rationalization," illustrated in the diagram above.

> The rationalization mechanism, on the other hand, has three methods of dealing with upsetting statements—each of which represents a different way of denying the psychological responsibility of the actor for the action.

> They are: by assigning prime responsibility for the actor to another actor who controls the original actor; by assuming the original action was an unintended consequence of some other action truly intended by the actor; by assuming that the original action will set other events in motion ultimately leading to a more appropriate outcome. (Wardrip-Fruin, 2009)

Many subsequent AI systems reference the Goldwater Machine as an early example of computerized storytelling. Subsequent systems, like Jaime Carbonell's POLITICS and Michael Mateas's Terminal Time, would be based off of this work. Mateas summarized the Goldwater Machine's functions, as follows: "The Goldwater Machine mimicked the responses of conservative presidential candidate Barry Goldwater to questions about the Cold War." (Mateas, 2000) In this section, we gave an example of what it looks like to model an AI after human intelligence, in regards to belief systems and rationalizations. What would an interactive story experience look like in applying similar theories? We discuss Terminal Time in the next section.

5.5 Terminal Time: Combining Reactive Believability with Models of Belief Systems

Terminal Time was an installation piece that used ideological goal trees to retell a period of history. If we only look at story intelligence as a composition of events, we take for granted the meta-system of coherently tying together the belief-systems of the audience and their perception of the characters and context. With the Goldwater Machine, Abelson deepened the believability representations of chatbots, like Parry and Eliza, through modeling bias and human interpretation as formally defined belief-systems. By applying story intelligence to historical documentaries, both the believable reactions to human input (like Eliza) and the adaptability to belief systems (like Goldwater Machine) can be observed through the resulting variations of fixed events over a meaningful length of time (rather than case-by-case). Particularly for Terminal Time, depending on the circumstance of the screening, each audience experienced a very different retelling of what is constrained by events in our history.

Terminal Time's AI architecture was based on three components: knowledge base, ideological goal trees, and story experts. (Mateas et al., 1999)

- The knowledge base is a vast knowledge web
- Ideological goal trees choose historical events in accordance with viewer responses.
- Story experts utilize narrative conventions to plan, compose and evaluate final story texts.

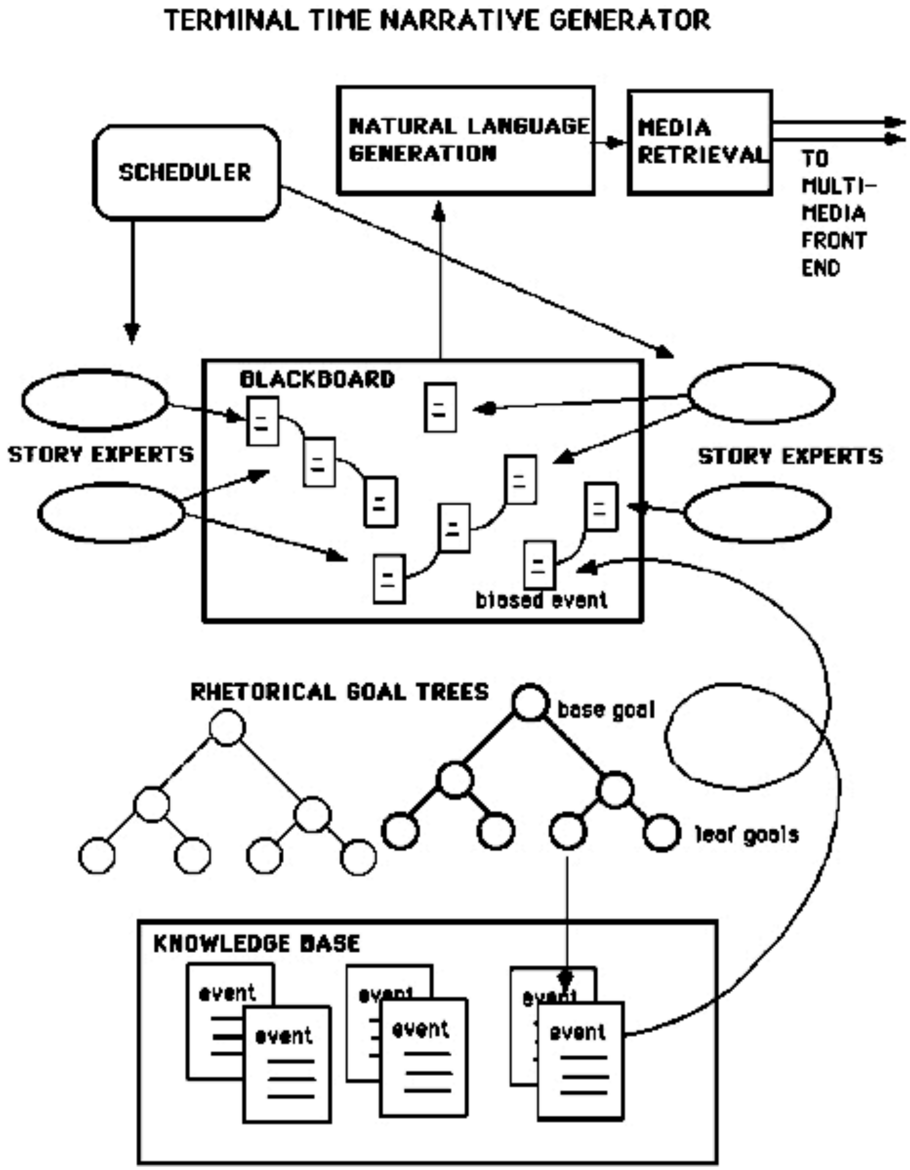

Figure 5-7. Terminal Time System Diagram.

The story of Terminal Time is represented by the designated time periods shown on the timeline below, starting from 1000 AD to the early 2000's. The presentation is directed by the audience interaction and determined by the ideological goal trees.

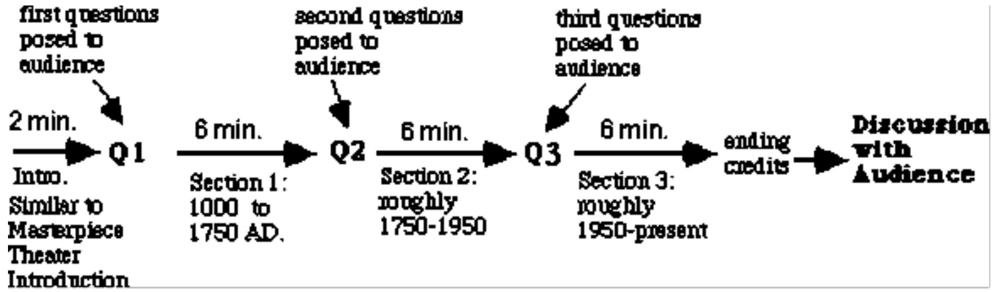

Figure 5-8. Timeline of content and audience interaction.

Terminal Time, like subsequent systems such as Nick Montfort's Curveship, deemphasized the causal models for believable interactions (like we saw with Eliza), looking beyond believable events towards underlying beliefs (like we saw with Goldwater Machine), and focused on aspects such as ideology, rhetoric and presentation (Montfort, 2009). In these systems, we manage the variations of user experiences using what we will describe as supplementary variations.

In this chapter, we discussed Intelligent Narratives from the historical perspectives in AI. We looked at three particular use-cases, chatbots (Eliza and Parry), Goldwater Machine, and Terminal Time. Although there are many ways to represent and formalize narrative intelligence, many of the early story generation work focused on composing characters and events. The examples discussed in this chapter highlight aspects of narrative intelligence that focus more on the beliefs and interpretation. This benefits from a different framework of in addition to discourse events and ontological modes of play that will be discussed in Chapter 7. To explore this more deeply, we will describe the evaluation process and results for a fully implemented AI system, Declarative Optimization-Based Drama Management (DODM). This study lays the motivation for Authorial Leverage in Chapter 7.

Chapter 6 – Declarative Optimization Based Drama Management (DODM)

To dive deep into an AI driven story experience, we implemented Declarative Optimization-Based Drama Management (DODM) for a Zelda-like dungeon world. This system was built towards alleviating the authorial burden that interactivity introduces. In this project, the primary focus was on forming satisfying story experiences through a causal model of storytelling with Artificial Intelligence. The use of AI; however, contributes additional authorial burden that did not exist with traditional forms of story authoring. To identify the leverage gained through the use of this (or any) AI system, an authorial leverage evaluation metric was used to assess the gains and losses from this alternative form of designing interactive stories. Following lessons learned through drama management, a new and different approach was attempted through the RoleModel project (Chen et al., 2010), discussed in Chapter 8.

The formalized definition of Drama Management came out of the Oz project at Carnegie Mellon, highly informed by Brenda Laurel's work in interactive drama, as well as work by Mateas and collaborators at Georgia Tech. Nelson & Mateas built a search-based agent that listened for game events (or plot-points) to occur. This agent, or Drama Manager, turned storytelling into a search problem much like chess, seeking to create the best possible experience--much like how Deep Blue aimed to checkmate its opponent (Nelson et al., 2006). DODM underwent various types of evaluation, resulting in a generalized theory of Authorial Leverage (Nelson & Isbell, 2008; Chen et al., 2009).

In this work, we focus on DODM, an approach to drama management based on plot points, DM actions, and an evaluation function (Weyhrauch, 1997). The two chapters we discuss two studies of DODM: (1) an audience-based evaluation outlined in this chapter, and (2) an author-based evaluation (Authorial Leverage) outlined in the next chapter. The studies show two different approaches to evaluating the expressivity of DODM.

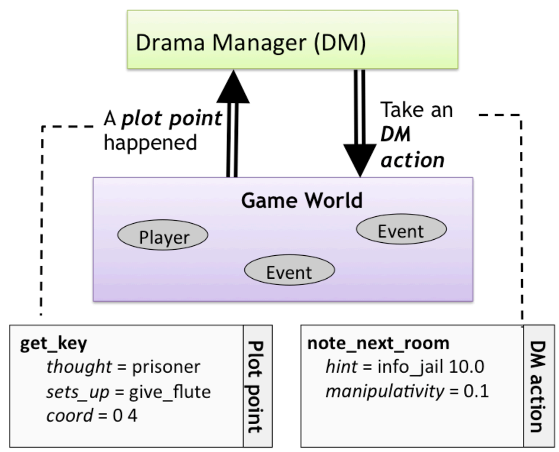

Figure 6-1. Drama Manager Architecture.

Figure 6-1 above is the architecture diagram for the DODM AI System. Such a system was designed to help mitigate authorial burden; however, it also creates additional authorial challenges that did not exist in more traditional forms of authoring. So, how can we overcome this authorial obstacle in creating interactive stories? Before we can answer this, we have to understand what exactly is being authored or managed when we build interactive stories.

Plot points are important events that can occur in an experience. Different sequences of plot points define different player trajectories through games or story worlds. Examples of plot points include a player gaining story information or acquiring an important object. The plot points are annotated with ordering constraints that capture the physical limitations of the world, such as events in a locked room not being possible until the player gets the key. Plot points are also annotated with information such as where it happens, or what its subplots are (see Figure 6-2).

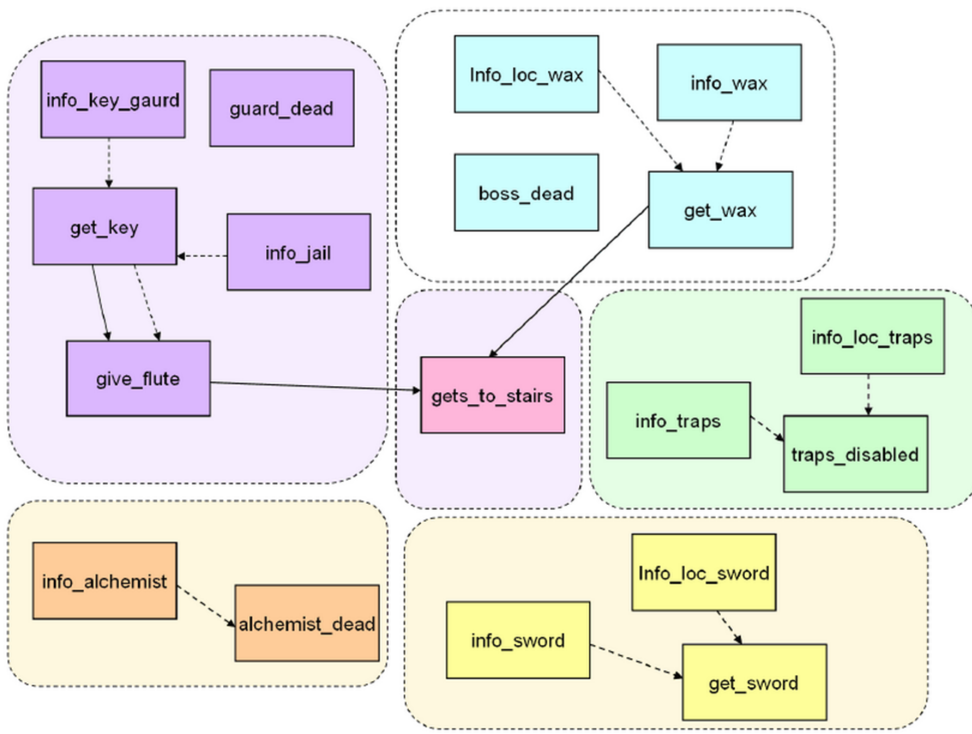

Figure 6-2. Plot points for the EMPath DODM System.

With DODM, the author can locally change the evaluation function resulting in global changes. The evaluation function, given a total sequence of plot points that occurred in the world, returns a "goodness" evaluation for that sequence. This evaluation is a specific, author-specified function that captures story or experience goodness for a specific world. While an author can create custom story features, the DODM framework provides a set of additive features that are commonly useful in defining evaluation functions (e.g. Weyhrauch, 1997; Nelson & Mateas, 2005). With such policy changes, we can observe the complexity through script-and-trigger equivalent, turning the operations that maximize evaluation scores to a tree like, if-then-else branching structure which we will discuss in the next chapter.

DM actions are actions the DM can take to intervene in the unfolding experience. Actions can cause specific plot points to occur, provide hints that make it more likely a plot point will occur, deny a plot point so it cannot take place, or un-deny a previously denied plot point. When DODM is connected to a concrete game world, the world informs the DM when the player has caused a plot point to occur. The DM then decides whether to take any actions, and tells the world to carry out that action.

Given this model, the DM's job is to choose actions (or no action at all) after the occurrence of every plot point so as to maximize the future goodness of the complete story. This optimization is performed using game-tree search in the space of plot points and DM actions, using expectimax to backup story evaluations from complete sequences.

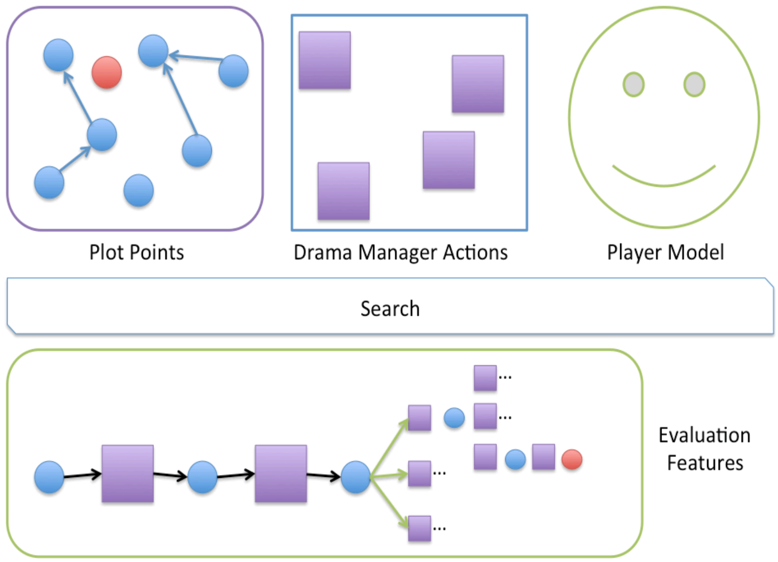

Figure 6-3. Drama Manager Components.

6.1 Drama Management System – EMPath

EMPath is the first real-time playable game that uses Declarative Optimization-Based Drama Management (DODM). DODM was originally proposed in 1992 in Peter Weyrauch's PhD Dissertation and implemented in 2001 with the Interactive Fiction, Anchorhead (Weyrauch 1997, Nelson 2006). EMPath was developed to work in conjunction with DODM in 2007 and took on the style of a 5×5 room dungeon game. In 2008, it was augmented from a 10-plot point game to an 18-plot point game with an 8×8 room dungeon. Overall, the user experiences were evaluated on over 100 users and assessed for contributions of authorial leverage through using methods in machine learning. (Sullivan, 2008).

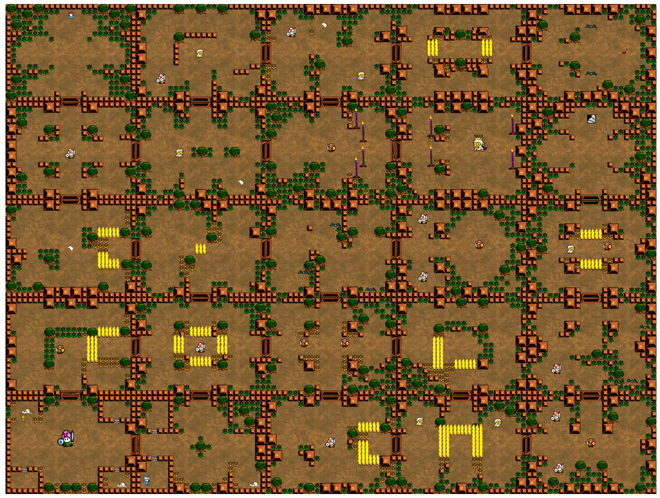

Figure 6-4. EMPath 1.0 World Map.

Given an authored world, the author is also required to provide:

- **Plot Points** – atomic units of dramatic events, initiated by the user. (e.g, gets_to_stairs and info_traps from Figure 6-2).
- **Drama Manager Actions** – actions available to the AI system, used to encourage the best possible experience. (e.g. deny_loc_candle and reenable_loc_candle).
- **Player Model** – a prediction model for a probabilistic distribution of what the user is likely to do next. (e.g. using the Manhattan distance to calculate the probability of a plot point being triggered).
- **Evaluation Features** – a set of functions that numerically define the qualities of preferred experience outcomes. (e.g. the story would be penalized from jumping between too many subquests, indicated by color in Figure 6-2).

Figure 6-5. When a drama manager action is triggered, the game world communicates it through story events, causing a plot point.

First, DODM takes a current state of the world, which comes in the form of a sequence of plot points and a list of valid drama manager actions that can influence the user. Then, DODM performs an exhaustive search to find all the total orders of possible completed story traces (from the point of the given state). As it searches, it uses the player model to assign a probability for the likelihood of outcomes and chooses the action (from the list) that is most likely to encourage the user to the best outcome. Evaluation features, defined by the author, are used to determine what makes one outcome better than another.

In a given trial run, the system has a list of plot points and their dependencies, illustrated in Figure 6-2. As soon as a plot point is triggered, the DODM evaluates all possible total orders from that point forward, shown in the bottom half of Figure 6-3. Intuitively, with every plot point triggered, the search space becomes smaller. DODM has operations called Drama Manager Actions, represented as squares in Figure 6-3. After each plot point, DODM can manipulate the world in response. The Drama Manager Action that yields the highest scoring branch of outcomes is chosen. Total orders are ranked and scored based off of author specified Evaluation Features. For example, in Figure

6-2, the plot points are color coded by topic. If the author prefers the topic to change as infrequently as possible, they can introduce a feature to increase the score of total orderings that have such outcomes. DODM makes use of Drama Manager Actions to influence the final ordering of plot points to maximize the score. In this case, to prevent the player from going off topic, DODM can deny entry into the undesired part of the map by locking the door. Player Models are used to simulate player behavior, and indicate how likely a plot point is to be triggered; typically, this is done through proximity (e.g., Manhattan Distance). One of the proposed advantages of DODM is that the author can locally change the Evaluation Features, resulting in global changes in the game.

6.2 User Evaluation for DODM

The game and system were evaluated by 100 users. Without knowledge of which group they were in, one group evaluated their experience with the drama manager, while the other evaluated their experience without the drama manager (or with the drama manager returning no action each turn). Results, however, were inconclusive.

As testers, a main challenge we faced was designing the experiences that we were evaluating. We asked the users, "Were you able to see a chain of game events, where one event caused another to happen?" We were looking for whether the authorially specified evaluation features would be noticeable to the user. Since evaluation features were applied to completed story-traces (a sequence of plot points that end with a terminating plot point), events that weren't modeled, such as death and the killing of minor enemies, were ambiguous. The main issue being, what is it that we were really managing?[7] What was defined as a "plot point" in the DODM system was indistinguishable from other game events, which makes the evaluation of the story's quality hard to measure. As a result, we developed a second study to identify the Authorial Leverage of DODM, shifting from a purely audience-based evaluation to one that considers the author's experience.

6.3 Designing AI Systems for the Author

From chat bots to propaganda generators to drama managers, these artificially intelligent systems are designed to achieve high-evaluating or believable outcomes (for their audience). Of course, with enough resources, time, and mind power, we could have more Façade-like user experiences; however, not as much work has been done to identify the challenges and bottlenecks. Evaluating the audience's experience has proven, in the case of EMPath, to be challenging, because the AI system may, in fact, be making many intelligent decisions, but the common user/player won't have the framework to detect, understand, or notice, especially if the refinement (or portrayal) of these interactions is mild. For another example, take the procedurally generated facial expressions of Trip and Grace (Mateas, 2005). It's not enough to just prove that users/audience would realize that something intelligent (or procedural) is happening; such dynamic behavior may prove not to be as meaningful or worth

[7] The plot points were grouped together with causal dependencies/preconditions. See diagram in appendix.

the cost of the effort necessary to design and build it. That's not to say that we don't care about designing new experiences, but that we should consider the cost of creating these experiences when designing and evaluating our AI.

There is a complex cost-benefit tradeoff to using AI for communicating and representing stories. In the past, systems like Façade were created to facilitate greater user agency. However, fewer studies advocate a shift in perspective and urge developers to balance user agency with author agency. AL (Authorial Leverage) is an author-centric evaluation model that compares the costs and benefits of using AI (or any sort of method/approach) to design a story experience. It considers the user-experience, but aims to quantify and model the author's experience.

The past evaluations of DODM thus far has established at least preliminary positive results for the technical features of optimization (Weyhrauch, 1997; Nelson et al, 2006; Nelson & Mateas, 2008); the effect on player experience (Sullivan, Chen, & Mateas, 2008); and the correspondence of some evaluation functions to expert notions of experience quality (Weyhrauch, 1997). None of this establishes the usefulness of DODM for authors, however, if similarly impressive results could have been achieved just as easily using traditional trigger-logic authoring techniques.

Traditionally, interactive story experiences are authored with sets of scripts and triggers: the author specifies particular events or world states that trigger scripts, which then perform some sequence of actions in response. One way to understand the operations of a DM is to generate script-and-trigger logic that acts the way the DM does. We do that by generating a large set of traces of the DM operating on a number of different stories, and then using a decision-tree learner to summarize the DM's operation. The internal nodes in the learned decision tree, which split on state values, correspond to the tests that exist in triggers; the leaves correspond to scripts to execute, represented by DM actions. A particular path from the root node to a leaf defines a script to execute, given the conjunction of the set of triggers along the path. This becomes the initial study towards measuring the Authorial Leverage of AI systems.

In the previous chapters, we discussed why there aren't more Façade-like interactive drama experiences. We looked at the conventional obstacles with building interactive stories, and discussed the AI systems that aimed to alleviate these challenges, satisfy believability, and execute intelligent interactions. In the case of DODM, and many similar studies, the user/audience's experience is the primary concern for determining success.

The next chapter proposes a more balanced analysis of storytelling between audience and author (Authorial Leverage) as a step forward in the right direction. The mark of success would be to see more Facade-like playable stories. Getting around the hang-ups with user evaluation and disambiguation of what the AI system is or isn't doing for DODM in EMPath, we also did a simulation-based evaluation centered on a metric for authorial leverage, which will be discussed in the next chapter.

Chapter 7 – Authorial Leverage

The drama manager (DM) monitors an interactive experience, such as a computer game, and intervenes to shape the global experience so that it satisfies the author's expressive goals without decreasing a player's interactive agency. Most research on drama management has proposed AI architectures and provided abstract evaluations of their effectiveness; a smaller body of work has also evaluated the effect of drama management on player experience. Little attention has been paid, however, to evaluating the authorial leverage provided by a drama-management architecture.

Authorial leverage is the power a tool gives an author to define a quality interactive experience in line with their goals, relative to the tool's authorial complexity. It has been pointed out that the "burden of authoring high quality dramatic experiences should not be increased because of the use of a drama manager" (Roberts & Isbell, 2008), but determining whether that is the case depends on determining both the complexity of an authoring approach and the gains it provides. This requires determining, for a given architecture, the additional non-linear story complexity a drama manager affords over traditional scripting methods.

We proposed three criteria for evaluating the authorial leverage of a DM: 1) the script-and-trigger complexity of the DM story policy; 2) the degree of policy change given changes to story elements; and 3) the average story branching factor for DM policies versus script-and-trigger policies for stories of equivalent quality. These criteria will be described in more detail below. We applied these criteria to declarative optimization-based drama management (DODM) by using decision tree learning to capture equivalent trigger logic, and show that DODM does in fact provide authorial leverage (Chen et al, 2009).

Earlier in Chapter 1 we gave an abstracted representation of Authorial Leverage:

$$\text{Authorial Leverage} = \frac{\text{Audience Experience}}{\text{Authorial Effort}}$$

Expanding on this representation, we define Audience Experience as a product of Quality, Variations, and Control. Conceptually we think of authorial leverage as follows:

$$\text{Authorial Leverage} = \frac{\text{Quality} \times \text{Variations} \times \text{Control}}{\text{Authorial Effort}}$$

In detail, these four properties are defined as follows:

Quality	This value is typically determined by user evaluation. If we can deliver a better experience without having to compromise viable variations and that costs the same amount (or less) in effort, we have created leverage. (**User focused**)
Variations	This value determines the diversity among potential experiences. In previous work, this has been done through comparing play traces. If we demonstrate an increase in legal variations of the same quality, or manage to create better sets of interesting variations without increasing effort, then we have created leverage. (**User focused**)
Control	If we are able to make changes, control and extend a story world, or create a brand new story world without compounding the effort or breaking the user-experience, then we have gained leverage. This value represents the precision and integrity of how well the audience-experience stays true to the integrity of the design or the authorial intention. If changes to the interactive space create nonsense or break the overall experience, then the system is inflexible. (**Author focused**)
Effort	We find the script-and-trigger policy (traditional approach) that produces quality experiences equal to the new approach. Then, within a similar space of interesting variations, we have a quantitative measure of effort. This is the amount of effort it would take an author to create the entire experience without an AI system. In practice, this was done by comparing the number of rules and specifications that are needed for a functioning or playable experience. (**Author focused**)

Table 7-1. Authorial Leverage

Technology can expand the possibilities of narrative both for those who experience and those who tell stories, in particular by making narrative interactive. Authoring interactive narratives, however, has proven to be quite challenging in practice. Although it shares some qualities with non-interactive storytelling, narrative in games delivers a highly interactive experience, which requires new ways of approaching authoring. Traditional approaches to authoring interactive stories in games involve a scripted and heavily linear process, and it is difficult to extend this process to large stories with complicated interactivity. Drama managers provide an alternative approach, by allowing the author to assume a system that knows something about how to manage the story at run-time. Such approaches, however, are difficult to evaluate from the perspective of the author.

Scholars have studied how drama management can improve experience quality. (Nelson, 2006) Improved user experience does not necessarily correlate with authorial benefit, however, one needs to prove that traditional authoring methods could not have achieved the same results, or that they would have required considerably more effort to do so.

7.1 Evaluating DM via equivalents

A way to get at that comparison is to look at the set of traditional trigger-logic rules that would be equivalent to what a drama manager is doing. We propose three criteria for evaluating the authorial leverage of drama managers in this manner: the equivalent script-and-trigger complexity of their policies; policy change complexity; and the average branching factor of their policies. We present preliminary work applying these metrics to declarative optimization-based drama management (DODM), by examining the equivalent trigger-logic for a drama-manager policy as captured by a decision-tree learner. In the next chapter, we measure the Authorial Leverage of DODM, looking at three perspectives on how a system like DODM provides leverage.

Complexity of script-and-trigger equivalents. First, if the script-and-trigger equivalent of a DM policy is unreasonably complex, then the DM is capturing a policy that would be infeasible to author as script-and-trigger logic. This would show that the DM provides authorial leverage compared to script-and-trigger logic. We can determine the smallest decision tree that achieves performance reasonably close to the drama manager, and qualitatively consider whether it would be reasonable to hand-author it. Alternately, we can start with a reasonable hand-authored policy for a small story world, and see how the complexity of required new additions scales as we add additional events and locations in the story world.

Ease of policy change. Second, if experiences can be tuned and altered easily by changing some DM parameters (e.g. the author decides the experience should be faster paced), and the equivalent changes in a trigger-logic equivalent would require many complicated edits throughout the system, then DM adds authorial leverage. DODM in particular uses a number of numerical values/weights/probabilities to define experience goals, which can be changed to re-weight criteria in decisions throughout the story. Other drama managers can allow for changes such as adding or removing story goals in a planning formalism. If simple changes at those levels of authorship result in a noticeably different script-and-trigger equivalent policy, the DM effectively allows an author to rescript the original from a compact representation, or to easily create a set of variations on a given experience.

Variability of experiences. The first two leverage metrics were based on the relationship between the amount of work and the quality of the work's outcome. A third measure of leverage is necessary to ensure that there are a variety of diverse experiences in addition to stories of great quality. If we only considered the first two metrics, an AI system that produces the same high quality experience every time would be considered to have significant leverage. It is necessary to consider frequency of variability because high quality stories are easily hand authored, although they are difficult to author in large numbers.

7.1.1 Decision Trees

We induced decision trees from example drama-managed story traces using the J48 algorithm implemented in Weka, a machine-learning software package. Each drama manager decision is made in the context of a partially completed story, so the training data is a set of (partial story, dm-action) pairs, generated by running the search-based drama manager to generate thousands of examples of its actions (done through using the Player Model as a simulated player, described in Chapter 6). Partial stories (the independent variable) are represented by a set of boolean flags. These indicate whether each plot point and DM action has happened thus far in the story, and, for each pair of plot points a and b, whether a preceded b if both happened. The tree that results can be interpreted as a script-and trigger system. Each interior node, which splits on one of the boolean attributes, is a test of a flag. The path from a root node to a leaf passes through a number of such flag tests, and their conjunction is the trigger that activates the script at the leaf node, represented by a DM action to take. The tree format is simply a compact (and inducible from data) representation of the total set of triggers. Decision trees of various sizes can be induced by varying the pruning parameters: a low degree of pruning will effectively memorize the training examples, while a high degree of pruning captures a small script-and-trigger system that accounts for as much of the DM's behavior as possible, given the small permitted tree.

Any of the policies—the actual DM policy or any of the decision trees—can be run with a simulated player to generate a histogram of how frequently experiences of various qualities occur. More successful drama management will increase the proportion of highly rated experiences and decrease the proportion of lower-rated experiences.

Varying the degree of pruning allows us to see how much performance is sacrificed by limiting to a simple script-and-trigger system; or alternately, to see what level of script-and-trigger complexity is needed to achieve performance similar to the drama manager.

7.1.2 DM policy evaluations in EMPath

We performed our preliminary evaluations on EMPath in Chapter 6, a Zelda-like adventure game (Sullivan, Chen, & Mateas, 2008) that was developed to test DODM in a traditional game genre (described in the previous chapter). It is set in a 25-room dungeon and has at most 10 plot points that can possibly occur. In addition to the game, there are 32 DM actions that DODM may choose to employ at various points in the story (33 DM actions when counting the choice to do nothing). Figure 7-1 below shows the world's 10 plot points and their required precedence relationships:

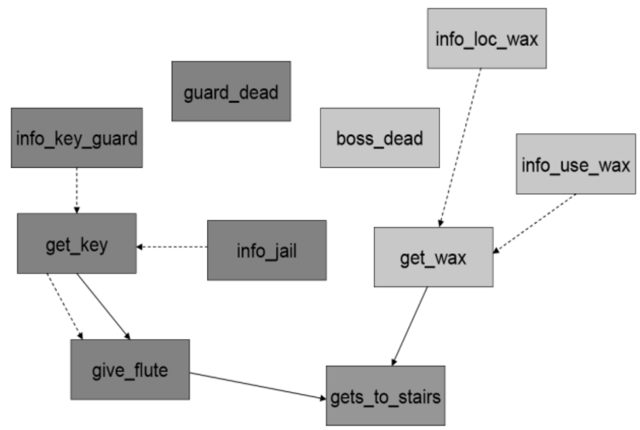

Figure 7-1. This is the directed-acyclic graph for the story world.

We ran DODM in this world with a simulated user to generate 2500 drama-managed story traces, producing 22,000 instances of training data from which to induce a decision tree. To vary pruning, we varied minimal terminal node size (or leaf node count), with a larger minimal terminal node size resulting in smaller trees as splitting does not continue as far.

The following histogram shows the performance of the drama manager in the EMPath story world, compared to the performance of a null policy (which always takes no DM actions) and a number of trees at various levels of pruning. It is apparent that the performance of the smallest trees (greatest pruning), such as the one labeled 1000, performs only slightly better than the null policy, whereas the best match with the search-based policy (the actual DM policy) is found at moderately low levels of pruning, labeled 200. In addition, the least-pruned trees (e.g. 50) overfit to the particular runs in the training set, as we'd expect, resulting in worse overall performance, so they aren't a good capture of an equivalent policy.

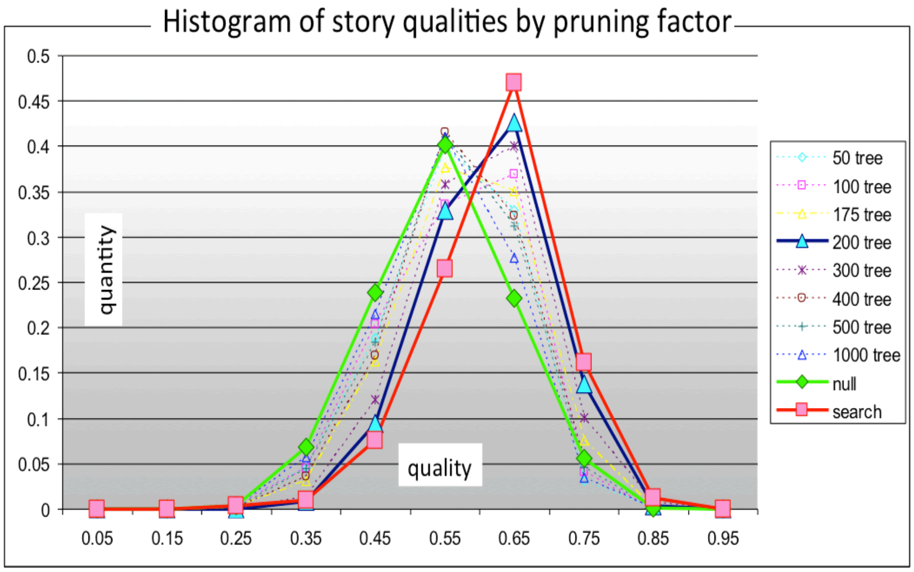

Figure 7-2. Histogram of story qualities given the search policy.

The tree trees below represent the highly pruned (1000) policy, and the best performing (200) policy, respectively (for a zoomed in, more readable version, see Figure 7-5):

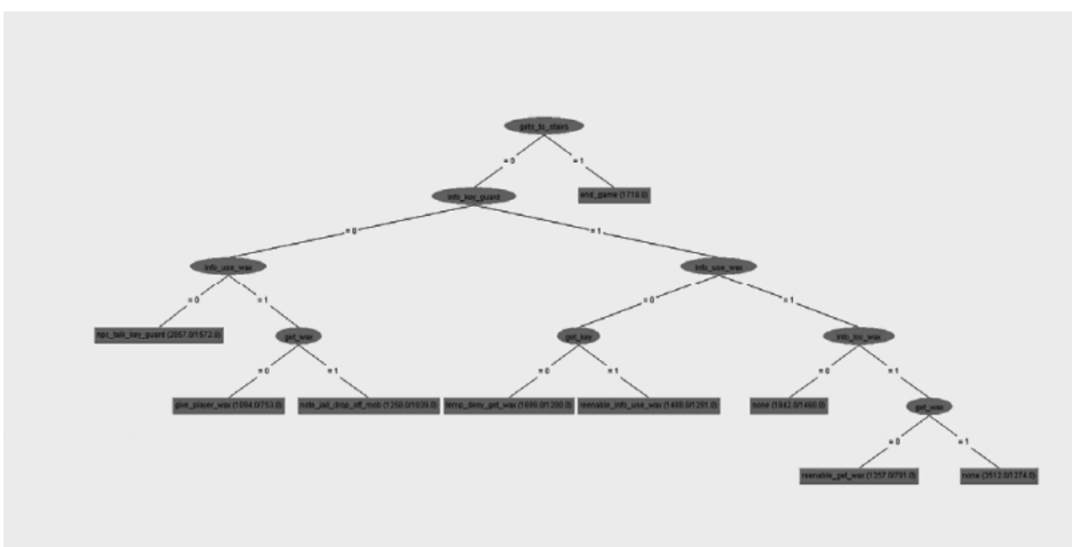

Figure 7-3. The poorly evaluating decision 1000-tree (17 nodes).

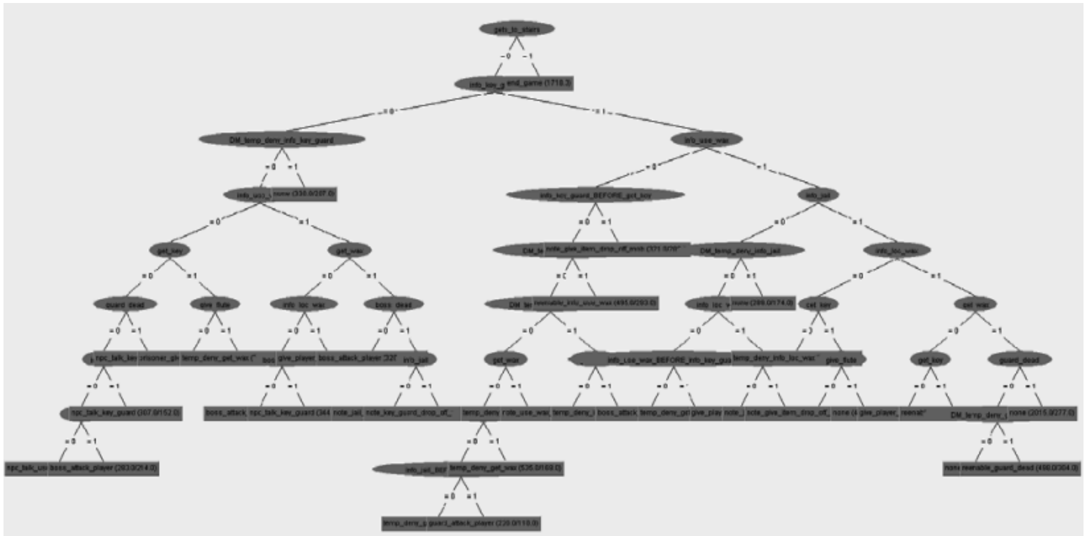

Figure 7-4. The highest evaluating decision 200-tree (70 nodes).

Although this zoomed-out view gives only a general idea of the policies, the second policy is already clearly quite complex for such a small story world, and the more reasonable first policy empirically doesn't perform as well.

As a result of using the player model to generate story-traces, DODM was able to easily produce thousands of stories along with consistent evaluations. These stores were used as instances in a decision tree learner and used to build if-then-else trees of varying pruning factors (or complexity). Finding the pruning factor that most closely matched the DODM, the complexity of the tree was used to determine a measure for the traditional authoring of interactive story. The zoomed-in view of part of the best-performing (200) tree below shows some of the equivalent script-and-trigger logic that it captures:

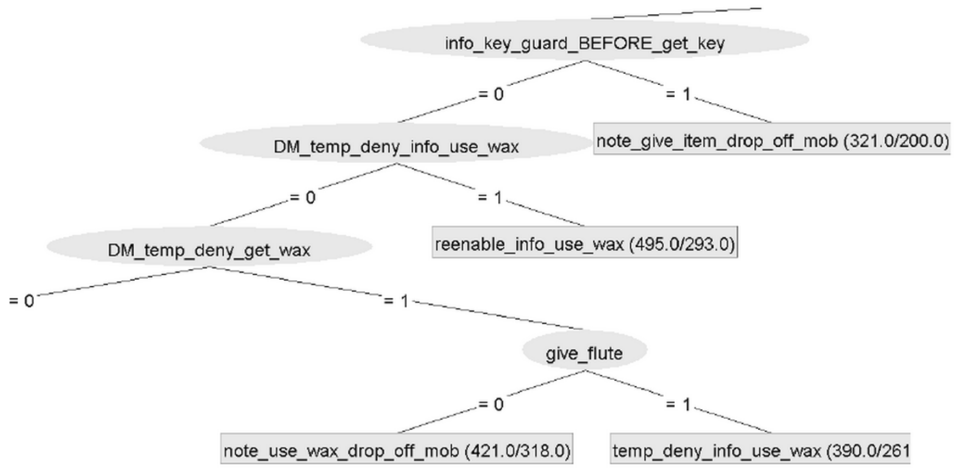

Figure 7-5. Part of a decision tree built from stories generated by DODM. Zoomed in view of the 200 pruned tree.

One trace through this segment specifies the following rule. If info_key_guard_BEFORE_get_key is false (i.e. either info_key_guard or get_key plot points haven't happened, or the info_key_guard plot point happened second); and the DM action temp_deny_info_use_wax has not been used; and the DM action temp_deny_wax has not been used; and the plot point give_flute has happened; all conjoined with any tests further up the tree; then take the DM action temp_deny_info_use_wax. This is specifying a series of exclusion tests, followed by a choice of what to do if all of them pass; that choice depends on whether the flute has been given yet. Hundreds of these sorts of rules are generated automatically. While they could all be authored by hand in principle, the fact that even such a small story world requires a tree of this size to reasonably approximate the DM's performance gives some indication of the infeasibility of doing so.

7.1.3 Ease of Expanding the EMPath world

As a way of testing the easy of policy change (the second authorial leverage criteria), we created three versions of EMPath with increasing world complexity. Table 1 summarizes the three game variants.

	# plot points	# DM actions	# quests	map size
empath-small	10	33	3	25
empath-med	14	47	5	64
empath-large	18	62	6	64

Table 7-2. Game policy variations.

Each story variation is used to create its own decision tree training data, by producing 1000 stories each. The training data is built from the partial stories from each 1000-story set. Story worlds that were bigger had larger data sets as a result (8780, 12594, and 16437 respectively).

Recall that the second authorial leverage criterion is ease of policy change. Using DODM, to incorporate the logic for the new subquests into the game, all the author has to do is provide the DM with the new plot points and DM actions, and include the larger world map in the player model (see [Sullivan, Chen and Mateas 2009] for details on the player model). To change the policy for the script-and-trigger-equivalent trees, the author would have to manually add and delete trigger conditions to account for the new content. Given our EMPath variants and the induced script-and-trigger equivalent logic, we need a way of comparing the differences between trees in order to measure the ease (or difficulty) of changing one tree into another. As a simple of measure of this, we find a decision tree that best fits the search-based DODM performance for each

EMPath variant (using the same techniques as described above), and compare the sizes of the trees. If the sizes of the trees vary significantly between EMPath variants, then there would be significant authorial difficulty in manually creating new script-and-trigger logic for each variant. Note that, even if the trees are the same size, there could be significant differences between trees, differences that would best be captured with some version of edit distance. But tree size gives us a first approximation of this difference.

Figure 7-6 below graphs the node size of the best-fitting decision tree for each of the variants. There is a significant increase in the node size of the decision tree from empath-small to empath-med and from med to large. In this instance, it is clear that the number of changes is noticeably different.

From these histograms, the optimal histogram can be selected and analyzed for its complexity. Tree sizes grow significantly from empath-small to empathy-large. This means that, to expand the game from empath-small to empath-large, requires hundreds of edits to the script-and-trigger-equivalent logic.

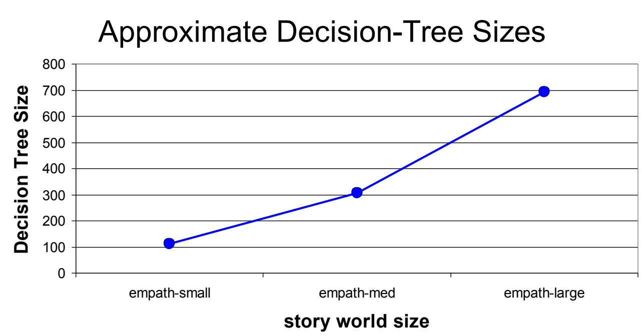

Figure 7-6. Approximated complexity for the most optimal decision tree policy.

To determine if, using the second criterion, DODM provides authorial leverage, we need to compare these hundreds of edits with the authoring work required using DODM. To include 8 additional plot points and 29 drama manager actions, the author must describe each plot point and action to the DM. Plot points and DM actions are defined by a list of attribute/value pairs.

Consider the get_sword plot point as an example of one of the 8 new plot points added to expand from empath-small to empath-large.

- get_sword
- QUEST = sword
- MOTIVATED_BY = {info_sword, info_loc_sword}
- COORD = 6 0

The quest attribute describes which quest the plot point is part of, the motivated_by attribute describes the list of plot points that should motivate, for the player, this plot point happening, while the coord attribute stores the initial map location at which this plot point will occur (initial location of the sword, which can potentially be moved around by drama management actions). When evaluating the quality of potential future sequences of plot points, the evaluation function will use the attribute values to determine the quality of a particular sequence; for example, the evaluation function would decrease the rating of a sequence in which info_sword and info_loc_sword don't happen before get_sword, because the player acquiring the sword is not motivated in that sequence.

Now consider give_player_sword, one of the 29 drama management actions added to expand empath_small to empath_large.

- give_player_sword
- CAUSES = get_sword
- MANIPULATION = 0.9

This DM action can force the plot point get_sword to happen by making an NPC walk up and give the sword to the player (with appropriate dialog from the NPC). The manipulation attribute indicates how manipulative the player is likely to find this action (how rail-roaded the action might make them feel). The value of 0.9 (1.0 is maximum) indicates that this is a strongly manipulative action.

In addition to defining plot points and drama manager actions, the author also defines an evaluation function, expressed as a linear weighted sum of evaluation features. An example of an evaluation feature is one that scores how motivated the events in a plot point sequence are, that is, how often, for each plot point in the sequence, its motivated_by plot points happen earlier in the sequence). The author can tune the relative importance of the different features by adjusting the weights associated with each feature. Adjusting the weights of the evaluation features, determines characteristics for the overall quality metric of the story. So, even without adding any additional plot points or DM moves, the author can adjust the experience purely by changing evaluation features or adjusting weights. Thus, another way to measure ease of policy change would be to learn decision trees for several different weightings and evaluation feature combinations, and measure how different these trees are from each other. In this paper, we only address policy change associated with adding new plot points and DM moves.

7.1.4 Variability of Stories for EMPath

The final measure for authorial leverage is in the variety of quality experiences. The simplest way to measure variety is to sum up the total of unique stories. Figure 7-7 below shows the histogram for number of unique stories (out of 50,000 simulated player runs) in the empath-small story world for trees of decreasing size, where the leftmost tree is the best fitting tree. The first thing to note is that the tree that best matches the DODM policy, the 137 tree, still produces over 6000 unique stories (unique sequences of plot points). Thus, DODM is not forcing a small number of stories to always occur. Second, note that as we move towards smaller trees (increased generalization), the number of unique stories grows (more than 14000 in the smallest tree). But we know from Figure 7-3 that smaller trees result in worse story quality histograms. Thus, the higher script-and-trigger complexity of the larger tree (the DM-equivalent tree) is producing an increase in story quality while still supporting a wide-variety of experiences.

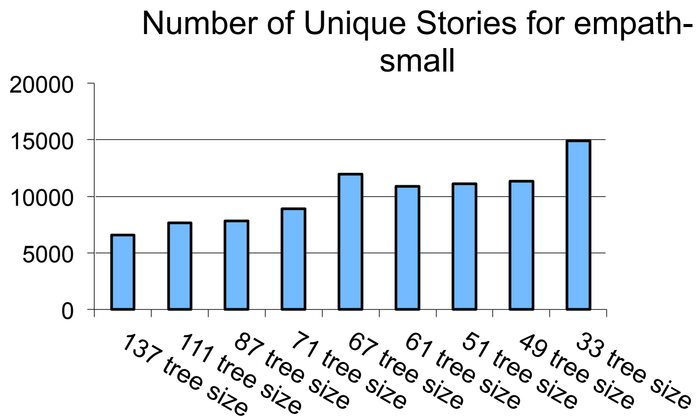

Figure 7-7. Histogram for unique stories according to tree size.

7.2 Decision-tree policy issues

Although decision trees are a nice way of automatically capturing a form of a DM policy that can be interpreted as a script-and-trigger system, there are a few difficulties with the policies they produce. The generalization that takes place in decision-tree induction can produce choices of actions that would not be permitted in a particular state: the decision-tree-learning algorithm has no notion of the internal structure of DODM so it may make unsafe generalizations. Two instances where the decision trees produced invalid choices of DM action were: 1) taking causer DM actions that cause plot points which already happened; and 2) not knowing that denier actions for critical plot points must be re-enabled eventually. These are, in effect, uncaptured additional complexities in a correct DM policy that a script-and-trigger system would need to deal with. An improvement to the decision-tree induction that might capture them would be to produce a number of negative examples of such disallowed choices of DM actions, and use a decision-tree induction algorithm that allows negative class examples.

We propose that a major open issue in the evaluation of drama managers is their authorial leverage: the degree of authorial control they provide over an interactive experience as compared to the complexity of that authoring. Authoring drama-manager-like interaction in stories is commonly done via scripts and triggers. Therefore, one way a drama manager evaluates authorial leverage is by using decision trees to induce and examine a script-and-trigger equivalent form of a drama manager's policy. We proposed three methods for running the comparison: 1) consider the complexity and complexity scaling with story size of the script-and-trigger versions; 2) consider the ease with which stories can be rebalanced or changed by changing DM parameters versus editing a set of scripts and triggers; and 3) examine the variability of stories produced by a script-and-trigger system and a DM policy, e.g. by their branching factors.

7.3 Authorial Leverage Informed Storytelling

We proposed that a major open issue in the evaluation of drama managers is their authorial leverage: the degree of authorial control they provide over an interactive experience as compared to the complexity of the authoring involved. Since authoring drama-manager-like interaction in stories is commonly done via scripts and triggers, we proposed that one way to evaluate the authorial leverage a drama manager gives is to use decision trees to induce and examine a script-and-trigger equivalent form of a drama manager's policy.

We proposed three criteria with which to do the comparison: 1) examine the complexity the induced script-and-trigger representation; 2) consider the ease with which stories can be rebalanced or changed by changing DM parameters versus editing scripts and triggers (in this paper, the changes studied involve scaling storyworlds); and 3) examine the variability of stories produced by a script-and-trigger system and a DM policy, e.g. the implied branching factor of the experience.

We presented results in inducing a script-and-trigger equivalent form of a DODM policy in a Zelda-like world, EMPath, and evaluated it by our first proposed criterion, showing that the resulting policies are quite complex to hand-author even in this small domain. Secondly, we showed three versions of EMPath that vary in size, and measured how the decision tree equivalents scaled with these changes. This showed that adding a few plot points to the story world had drastic increases in decision tree complexities. Finally, we showed that using DODM leads to simultaneously higher quality and lots of variation from examining the variety and frequency of unique stories in conjunction with their story quality evaluations.

Three primary directions that future work should take are: evaluating other systems, developing further ways for investigating the performance measures, and making use of the learned script-and-trigger systems. The evaluation measures will need to be applied to other story systems in several story worlds, and ideally, would also compare DODM to other drama-management approaches using a similar evaluation of authorial leverage. The three approaches we took to evaluate DODM can be further refined; for instance, performing a more rigorous statistical analysis or implementing average-branching-factor to measure story variation. In addition to evaluating the authorial leverage of drama management, the script-and-trigger systems demonstrated that decision tree policies were drastically faster at run time, although building the trees may take weeks to preprocess. Future work should examine how these learned script-and-trigger policies can be used at runtime as a "compiled" version of the optimization-based drama manager.

It has been theorized that computers can lighten the authorial burden of digital storytelling using practices like story generation and drama management (Mateas, 2001). In practice, however, computers not have proven to be effective at directing stories, in part because they read and understand them poorly. In experiences like Benmergui's Storyteller[8], for example, a constraint evaluator makes a fine judge of story, but by relying on a very careful balance between story complexity, representation, and how constraints are expressed. We explore the practice of computational storytelling further through RoleModel.

In the next study, the goal is to discover ways to lighten the burden for authors and make computers better storytellers. Out of many possible solutions, we propose a declarative-rhetorical model that combines four well-established

---

[8] http://www.storyteller-game.com/

story-generation paradigms: author-models, world-models, story-models, and reader-models (Bailey, 1999).

In this chapter, we describe how AL provides a basic framework for measuring greater expressivity of a story system architecture. We formally introduce Authorial Leverage (AL) and illustrate the difference between an audience-centric evaluation versus author-centric. In particular, we re-evaluate DODM from the AL framework.

In the next chapter, we show how storytelling is a constraint satisfaction problem and outline the design process of this approach. As a result, this work also presents a case for operationalizing storytelling. In addition, we present criteria for evaluating RoleModel, and discuss how it sets the stage for future work in generative storytelling. The RoleModel study has three goals: (1) reducing the authorial burden of generative storytelling, (2) operationalizing storytelling to enable more computational leverage, and (3) specifically enabling interactive stories and story-generators to produce compelling story-variations, given fixed assets.

Chapter 8 – RoleModel Background

> [People] are thus forced, in some sense, to make their story acceptable and easily comprehensible both by their initial attempts to understand the events themselves and by their prior attempts to tell others their story. (Schank, 1990) - AI Researcher, Roger Schank, points out that the storytelling process is calculated and purposeful.

So far, we have looked at how computers tell stories and what their limitations are, and we have proposed a new way to think about the problem. Instead of taking the traditional content-and-discourse approach, our study takes a constituent and supplementary approach. Rather than focusing on the structure of the narrative from the audience perspective, our storytelling system, RoleModel, is designed to consider the authorial perspective as well. This chapter describes the background, motivation, and framework for RoleModel.

RoleModel uses constraint satisfaction for narrative reasoning. Logical models of story are a means for both story understanding and story generation. Declarative programming allows stageless representation of the structures important to storytelling, such as storyworld state, causality of events, character beliefs, and commonsense interpretations. Employing logical constraints to model a space of stories, we use the RoleModel system to generate variations for an online storytelling experience called Robot Detective. The story is introduced through a robot detective that generates variations of a particular crime, modeled after the Rashomon effect and Abelson's rationalization mechanisms. Robot Detective is the field's first demonstration of: (1) how RoleModel can be used as a generative storytelling system, (2) the stories generated by RoleModel, and (3) a fixed-asset approach to story generation through constraint satisfaction.

First, we discuss the AI traditions that inspire and inform the implementation of RoleModel. Second, we identify objectives in generative storytelling. Finally, we define and discuss constraint satisfaction as it applies to storytelling and story generation for games.

The main inspiration for this project comes from Robert Abelson's work on models of human affect and Kurosawa's film Rashomon (Abelson, 1963; Kurosawa, 1950). Our work is contextualized in the area of generative storytelling, or the use of AI for storytelling, and in particular, practices that rely on literary theory to model storytelling. The system is called RoleModel and focuses on establishing storytelling models through the benefits of the declarative programming paradigm.

8.1 RoleModel System

RoleModel is a story generator organized around explicit formal models of character roles. RoleModel expands the expressiveness of stories generated from arbitrary partial domain specifications by using a formal model of roles within an abductive logic programming framework. The system's authorial goals can be fully or partially specified as constraints in an abductive logic program. In particular, the RoleModel system focuses on representing and satisfying the role constraints of the story characters. Being able to specify the constituent aspects of the story allows the supplementary variations to be meaningfully composed and tuned. The remainder of this section will show example output from the system to demonstrate the authorial expressiveness enabled by a “stageless” abductive logic approach to story generation (Chen et al., 2010).

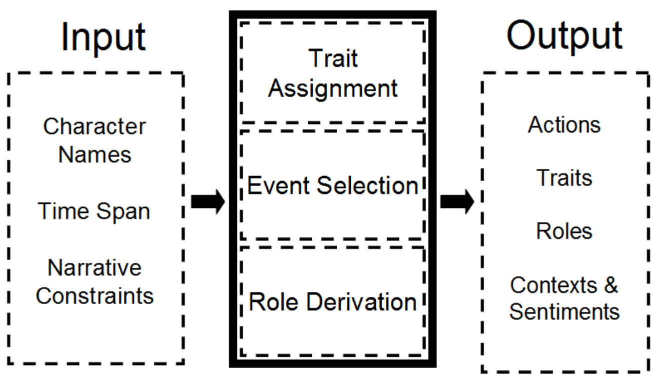

Figure 8-1. RoleModel input and output pipeline.

Character roles and archetypes play an important part in storytelling by providing motivations for character actions and introducing clearly recognizable dramatic interactions. Expert storytellers exploit character roles and role changing situations to manipulate users’ beliefs and expectations to bring about dramatic conflicts and resolutions. For example, in Kurosawa’s Rashomon, several re-tellings of a dramatic situation are presented to the viewer. In each narration, given from a different character’s point of view, the roles of the participating actors (e.g. Victim, Aggressor, Moderator) are manipulated to create coherent variations of the situation. Specific roles provide affordances for characters to undertake particular types of actions within the story. For example, in Rashomon, the woman's alternative roles of aggressor or victim provide the author with an option to create interesting variations on the aggressive episodes within the story. In intelligent storytelling systems, a rich formal model of roles enables authors to partially specify the domain and character constraints without sacrificing consistency of character behaviors with respect to their roles.

RoleModel is a story generator that explicitly models roles to generate meaningful variations of story situations. Due to the complexity involved in authoring complete and consistent formal domains that generate an authorially desired story space, we investigate the use of abductive logic programming to create models of possible story variations from a partially specified domain. Such a system provides authors with the ability to explore the space of possible variations given varying levels of story constraints.

In making roles a first class problem, our system took advantage of the strong perception of affordances for roles, such as victim or hero, in story. With a dynamic constraint space designed around maintaining roles, there are three authorial use-cases that can be effectively implemented: (1) a tabula rasa generator, which takes few or no constraints and autonomously generates varied narratives from the background theory; (2) a partially constrained generator, with which the author can specify additional story constraints (such as role fillers, character traits, and even the appearance of specific events within the story) on top of the background theory, without locking down a specific linear sequence of events, and (3) a highly constrained generator, with which an author can specify a linear story on which the system generates variations. In focusing on satisfying role constraints, the overall space of constraints can be viewed as properties of characters or properties of actions. Character properties include roles, traits, dynamic attributes, and sentiments towards actions, while action properties include a variety of contextual properties and causal constraints. The relationships between these constraints provide the background theory for the solver to use. For our prototype, generation involves asking the system to satisfy a list of additional story constraints (including an option of no constraints) on top of the background theory. The system produces a collection of grounded predicates (an answer set), where each collection corresponds to a concrete story that satisfies the constraints given the background theory. In the prototype, actions are represented using the event calculus, supporting temporal inferences about actions.

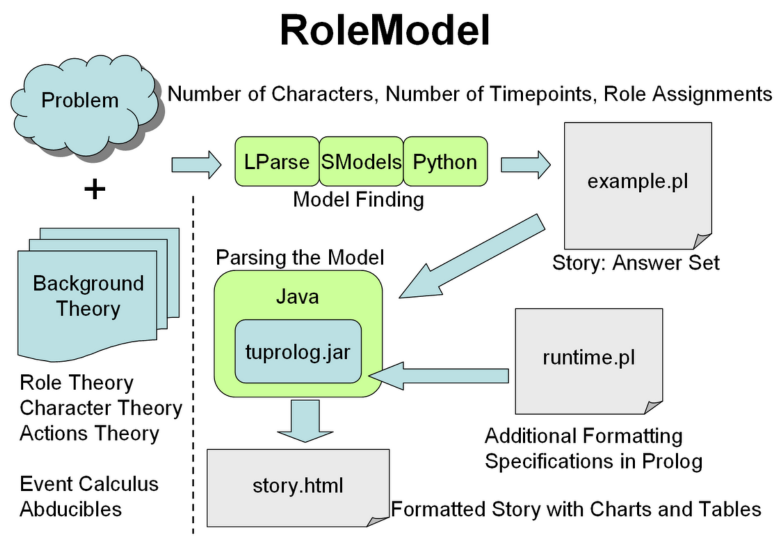

Figure 8-2. RoleModel System Architecture.

Based on given characters, role constraints, and goals, the RoleModel system aims to elaborate upon the given initial story assertions (or problem constraints) to establish or amplify character roles by adding preconditions on actions that fulfill character role constraints. In changing the role assignments, the system manipulates background knowledge and elaborates upon the story without breaking the author's specified initial story conditions. While many story generators emphasize the means by which they maintain causal consistency between actions, we were interested in how dynamic role assignment, and the implications that follow from roles, can be used to reframe similar action sequences to have very different meanings.

The author provides a file, called the problem specification, which contains the desired authorial constraints. This problem spec is combined with background theory on the particular desired story and passes through a constraint solver. The solver returns a story in the form of an answer set and is then passed through a parser to be visualized in an HTML file.

role assignments

	billy	matt	wendy
aggressor		X	
bystander			
victim	X	X	

trait assignments

	billy	matt	wendy
insane	X	X	X
sane			
strong		X	
trickster			

billy attributes

	0	1	2	3	4	5
alive	X	X	X	X	X	X
manipulated						
owned_by(billy)						
owned_by(matt)						
owned_by(wendy)						
restrained		X	X	X	X	X

Figure 8-3. HTML visualization of story answer set role assignment

```
% INPUT
person(alice).
person(bob).
person(eve).

t(0..3).

forbidden_role(aggressor,alice).
forbidden_role(aggressor,bob).
required_role(aggressor,eve).

required_action(alice,comforts,bob).
forbidden_action(eve,kills,eve).
required_action(bob,kills,alice).

% OUTPUT
happens(eve,tricks,bob,0).
happens(bob,ties_up,alice,1).
happens(alice,comforts,bob,2).
happens(bob,kills,alice,3).

sentiment(eve,tricks,bob,0,desire).
sentiment(bob,ties_up,alice,1,desire).
sentiment(alice,comforts,bob,2,regret).
sentiment(bob,kills,alice,3,desire).

has_trait(trickster,eve).
has_trait(trickster,alice).
has_role(victim,bob).
has_role(victim,alice).
has_role(aggressor,eve).
```

Figure 8-4. Example code.

In Figure 8-4 above, the author names three characters, chooses a span of four time points, and designates role and action constraints. Since the input gives an unordered list of actions, the system will build the required actions into the timeline. Incidentally, in finding a model that satisfies the authorial constraints, the randomly chosen model also determined that both bob and alice are victims (which was not specified nor precluded by the author). Despite what happens to a character or what a character does, roles need to be reasonably confirmed or denied. For the prototype implementation of RoleModel, negating aggression is accomplished by either showing regret or showing that the alleged aggressor was actually tricked into causing harm. In the example from Figure 8-4, although he did intend to commit murder, bob clearly was tricked was tricked by eve, satisfying the precondition of forbidden-aggressor by deferring blame to the "trickster," eve.

Supplementary variation will be managed through a formal representation of character role rationalizations. Belief systems will be lists of rules that require and forbid predicate bindings of role assignments, primarily, but also of character traits and actions. These lists of beliefs can be assigned to characters in the story, to an outside audience belief system, or to a derived or predetermined authorial goal. As characters perform actions, actions will have properties, like "harm," which create roles like "victim" and "aggressor." These roles can establish or nullify themselves with rules that rationalize what has happened.

By using roles as the parameters for belief systems, we are not defining a complete model for supplementary variation or proposing a model for belief systems; rather, we are following Abelson's observation: When an individual is presented with a challenge to his belief system, the problem he must solve is, "What am I to believe now?" By creating role-driven variations, the system aims to generate convincing narratives for all possible belief systems—a belief system being a set of character role assignments.

Similar to Abelson's affective models (Abelson, 1963), we pursue rhetoric and bias as the foundational parameters for variation. This is in contrast to the more popular mechanization-of-plot approach that most storytelling systems adopt (e.g., the drama manager in Façade (Mateas, 2001)). To understand and evaluate generative storytelling systems, we borrow terminology defined through work done in Authorial-Leverage.

8.1.1 Authorial Leverage

We use Authorial Leverage to understand the benefits and challenges that technology presents for authoring stories in generative systems (Chen et al., 2009). This theory of evaluation is represented in the conceptual equation below:

$$\text{Leverage} = \frac{\text{Quality} \times \text{Variations} \times \text{Control}}{\text{Effort}}$$

Quality	a measure of quality in user experience
Variations	a measure of significant and meaningful variations
Control	the amount of control the system or approach offers for changing & adding content
Effort	the cost of building content for a system or approach

Table 8-1. General definitions for Authorial Leverage

According to Table 8-1 above, enhancing the Authorial Leverage of a system by improving the user experience requires better quality and better variations. To improve the author experience, we require more authorial flexibility and a reduction in effort.

Traditional forms of authoring provide the most control in developing a quality story. In a linear story, the author has a high amount of control, but if she desires variations in the stories, this requires compounding effort. Looking at previous systems, there are various approaches with various tradeoffs. For examples, the EMPath system had inconsistently rated quality, took high amounts of effort, and provided moderate amounts of flexibility and control (Sullivan et al., 2008). A system like Minstrel requires substantial effort to design, and its quality is plagued by story nonsense. (Turner, 1993). The more complex a system becomes, the more effort is required to manage nonsense and incoherency. These projects, however, are progressing towards an ideal AI system. Table 8-2 below compares the Authorial Leverage measures of linear and branching stories. The last column indicates the ideal system and the penultimate column gives a (very) broad generalization of how artificial intelligence (AI) systems currently perform: these approaches are inconsistent in quality and flexibility.

	Linear Story	**Branching Story**	**Current AI Approaches**	**Ideal AI System**
Quality	High	High	Low-High	High
Variations	Low	High	High	High
Control	High	Low	Low-High	High
Effort	Low	High	High	Low

Table 8-2. An in-general overview of storytelling models.

The proposed RoleModel approach aims to achieve the high and low values in the Ideal System column above, which will be further explained in the discussion section of this chapter. We introduce the term Mechanized-Plot Model to refer to objectives in storytelling AI as plot generation and management. Façade and Prom Week are examples of mechanized-plot (Mateas & Stern, 2005; McCoy et al., 2011), while a system like Curveship would be mechanized-discourse (Montfort, 2009). Minstrel is an example of a mechanized-plot story generator (Turner, 1993), while Cambot is an example of mechanized-discourse generation (via cinematic visualization) (Riedl et al., 2008). RoleModel is a combination of these two approaches. Its research objectives are shown in Table 8-3 below. To understand research objectives, we designate the first column below the, more traditional, Mechanized-Plot Model (driven by plot) and compare it side-by-side with the goals of the Declarative-Rhetorical Model (driven by rhetoric, beliefs, and ideologies).

	Story/Discourse	**RoleModel**
Quality	Construct an AI system that approaches human-author quality	Maintain the Linear-Story level of (human-author) quality
Variations	Create the largest space of stories that meets the standard specified in the above quality measure	Create a controlled space of variations that maintain the validity of the author's specifications, navigated via Rhetorical Parameters
Control	This measure is less of a concern when designing user-end experiences, more of a concern for author tools	Develop reusable abstractions and rulesets for story
Effort	Reduce the authoring burden by enabling stories to be generated or adapted by the system	Construct an AI system that approaches the Linear-Story level of low effort

Table 8-3. Comparing our new approach, Declarative-Rhetorical, with an older approach.

Of these four measures, RoleModel's primary theoretical contribution is in how it manages its story variations. The rhetorical parameters will be defined in detail in the next sections of this paper.

8.2 Rhetorical Variations

In addition to modeling plot and discourse, our approach is inspired by rhetoric-driven AI pursuits. For example, we are interested in systems like Terminal Time and Politics because of their emphasis on ideological models (Mateas et al., 1999; Carbonell, 1978). Abelson's and Schank's theoretical work, which use models of human bias to create spaces of rhetorical variations, also informs our approach (Abelson, 1979; Schank 1990).

8.2.1 Hot Cognition

To model human affect, Abelson (Abelson, 1963) designed a program around belief, bias, and ideologies. He used the following definitions:

Belief	a sentence recoverably stored within an element
Belief System	a set of belief-calling elements which are themselves interrelated in a set of sentences
Belief Dilemma	a situation in which the individual is confronted with the apparent necessity of changing one or more beliefs.

Table 8-4. Abelson's definitions for framing human affect as computational problem-solving.

Abelson described a specific type of problem which requires attitudinal problem-solving, where an individual's belief system is challenged, and they must resolve this conflict. In his paper on Hot Cognition, Abelson provided a number of mechanisms for creatively reconciling gaps and contradictions in a story:

Stop Thinking	removing particular sentences from thought
Denial	denying the truth value of the sentence
Rationalization	the acceptance of the truth value of the sentence, but somehow deflecting its evaluative implications
Differentiation	the creation of two elements to replace one element
Transcendence	a difficult higher-order mechanism
Bolstering	a side process of evaluative change which has the function of compensating for some of the damage done by the imbalance

Table 8-5. Abelson's mechanisms for maintaining human bias.

Early iterations of the RoleModel system used Abelson's rationalization mechanisms for driving story-generation. Such mechanizations are seen in other systems aimed at automated propaganda or human bias (Chen et al., 2010).

8.2.2 Terminal Time

To leverage human bias for storytelling, Terminal Time, a more recent system, used ideological goal trees to retell historical events. Inspired by the work of Abelson and Carbonell, Terminal Time retold the same time-period in history, creating variations according to the beliefs of its live audiences (Mateas et al., 1999). Depending on the circumstance of the screening, each audience experienced a different retelling within the constraints of historical facts. Terminal Time generated narration that sewed together a fixed set of documentary clips in conveying rhetorical variations.

8.2.3 Story Skeletons

In his book Tell Me a Story, AI researcher Roger Schank uses the idea of story skeletons to demonstrate what a deceptive process storytelling can be, even if a person has all their facts straight.

> A storyteller might be more accurately described as a story-fitter. Telling stories of our own lives, especially ones with high emotional impact, means attempting to fit events to a story that has already been told, a well-known story that others will easily understand. Story-fitting, then, is a kind of deceptive process, one that creates stories that are not always exactly true, that lie by omission. These lies, however are not necessarily intentional. (Schank, 1990).

By creating and filling in gaps, we rationalize the information given to us in order to arrive at a satisfying understanding of what occurred. Story skeletons provide frameworks for expressing a series of events from a particular point of view. In graphics, the point of view is defined by a camera viewpoint. In storytelling, belief systems provide the dramatic point of view that influences which story skeletons are selected and how events are fit to the skeleton. Like graphics, perspectives help define what a camera sees from that point of view. Stories create analogous experiences from dramatic perspectives, belief systems being the dramatic point-of-view.

Story skeletons become an important component in the declarative models for RoleModel. Specifically, by using this idea of abstraction, demonstrated in the following two examples from Shank's Tell Me a Story: Narrative Intelligence, RoleModel is able to construct narrative constraints that comply with patterns of character behaviors. This is a straightforward translation from narrative understanding to logical rules, which we use to formalize our declarative constraints.

```
SKELETON: understandable-tragedy
    •  actor pursues justifiable goal
    •  actor selects reasonable plan to achieve goal
    •  plan involves selection of correct action
    •  action taken has unintended and unanticipatable result
    •  innocent people are hurt by result
SKELETON: justifiability-of-self-defense
    •  first actor pursues unjustifiable goal
    •  first actor selects plan
    •  plan has intention of negative effect on second actor
    •  second actor is justified in selecting a goal
    •  second actor selects justifiable plan
    •  plan causes action to take place that harms first actor
```

Figure 8-5. Roger Schank's example story skeletons.

8.3 Logic

In practice, RoleModel reduces storytelling to a constraint satisfaction problem, specified by authorial constraints. RoleModel uses authorial specifications, one of the four Authorial-Leverage measures, to ensure high quality of story, while representing these constraints as logical predicates and maintaining validity through formalized story rulesets.

8.3.1 Constraint Satisfaction

In constraint satisfaction, a set of possible assignments for a set of variables defines a space of possibilities. Constraints declare which combinations of assignments are valid or invalid. For example, in a murder mystery scenario, we can associate each character with a variable that describes where they were at the time of the crime. Without constraints, there are many possible explanations for the murder. After adding the constraint that character

X was in the same room as character Y, or that no more than two characters could have been in a certain room, fewer plausible explanations remain. The Boolean satisfiability problem is a canonical problem in Copmuter Science, and there are a number of algorithms available for automatically finding solutions that meet all stated constraints (Russell & Norvig, 1995). When using solver software for constraint satisfaction problems (CSPs), a programmer expresses what they want the computer to find without specifying how to find it. We use this technology to model a space of stories under a number of interesting constraints. Specific solutions to our CSP correspond to concrete stories that we can expose to a reader.

8.3.2 Logic in Storytelling

The event calculus (Mueller, 2010) is a logical formalism about reasoning about possible states of the world and the effects of events over time. One facet of the event calculus is the commonsense law of inertia, which states that if an event causes a state of the world (a fluent) to hold, that state holds until another event explicitly changes it. Building on a vocabulary of events and fluents, we can express statements about natural events and intentioned character actions, along with the effects of those events and actions on the location of characters, ownership of items, and the content of character memories and sentiments. Taken as a body of constraints, the event calculus allows us to express CSPs in which solutions are constrained to follow commonsense laws of story interpretation.

In Bemergui's Storyteller[9], a unique narrative puzzle game, logical rules define the conditions under which one state of the world will follow from another. By specifying only that a story has a specific interpretation, Benmergui implicitly defines a space of concrete stories that meet that requirement. In effect, Storyteller asks the player to enact the work of a constraint solver, searching through the space of stories to find one that, when interpreted, matches the goal condition for the puzzle. For example, the player may have to create a story where "Eve breaks two hearts, but ends up heartbroken" by visually representing it as a three panel comic.

RoleModel uses a logical representation built on top of the event calculus, and it uses constraints on interpretations similar to Storyteller. Instead of asking the audience to invent a story, it gives them a tour through the multiple stories that match a given specification. RoleModel employs answer set programming (ASP) (Gebser et al., 2012), a logic programming paradigm that has been used for procedural content generation in games (Smith & Mateas, 2011), as the concrete representation for constraints.

For this dissertation, work done in building and evaluating EMPath and RoleModel are the primary technical contributions of research. In the next section we will discuss the implementation and study for the rhetoric-driven story generator RoleModel applied to a murder mystery story.

---

[9] http://www.storyteller-game.com/

Chapter 9 – RoleModel Study - ROBOT DETECTIVE

"Robot Detective" is an online, storybook-like narrative containing illustrated snapshots of plot events (see Figure 9-1 for an example). The user is shown a total of five images, whose captions are generated by the answer-set solver. The user-experience design is meant to evaluate and explore human understanding of image-caption variations. As mentioned in the previous section, this RoleModel system, similar to Terminal Time, uses context to give fixed assets different interpretations. The story experience is named after Roger Schank's theory on story skeletons. The narrative space is inspired by the Rashomon film.

9.1 Rashomon Effect

Akira Kurosawa's film (Kurosawa, 1950) features variations of a crime retold by primary and secondary characters. Each character's recollection is equally valid, and yet incompatible with one another. This gave rise to a fiction storytelling convention called the Rashomon effect[10], or a not-fully-trustworthy storyteller who is known as the unreliable narrator. This film's use of the unreliable narrator[11] creates the effect of trying to reconcile equally plausible, yet contradictory variations of the same story.

---

[10] https://en.wikipedia.org/wiki/Rashomon_effect

[11] https://en.wikipedia.org/wiki/Unreliable_narrator

In Kurosawa, a judge gathers the entire cast of characters to piece together events of a murder mystery. Each character shares his or her limited experience, while simultaneously vilifying and victimizing each other. Similarly, Robot Detective positions the user as the judge of a particular story variation, pieced together by the "Robot Detective." The captions are represented as English sentences. Since the story-experience is built off of RoleModel, role assignments are used as rhetorical parameters that direct the space of variations. As a constraint satisfaction problem, the authorial specifications become the storytelling constraints.

In Robot Detective, the author specifications are derived from the artist-drawn assets. The constraints are anchored to each image to ensure that the story generator has a sense of what variations are appropriate in a given time-step. These five images represent high-cost story assets, while anything that the computer generates will be considered low-cost.

9.2 RoleModel + Robot Detective

As defined in Chapter 8, RoleModel is a story generator of character roles. It was written as a logic program for which a solver would return valid answer-sets. RoleModel was able to generate sets of valid stories that satisfied required story constraints, where each answer represented the information of a single generated story. The system leveraged narrative expectations of roles to control spaces of story variations. For Robot Detective, we call these controls Rhetorical

Parameters, and we use the same character assignments of victim, aggressor, and bystander that RoleModel used (Chen et al, 2009).

The original work outlined three use-cases for RoleModel's declarative use of narrative formalisms: (1) a tabula rasa generator, (2) a partially constrained generator, and (3) a highly constrained generator. Based on given characters, role constraints, and goals, the RoleModel system elaborated upon the given initial story assertions (or problem constraints) to establish or amplify character roles. In changing the role assignments (constraints), the system manipulated background knowledge, creating variations without breaking the initial story conditions as specified by the author. While many story generators emphasize the means by which they maintain causal consistency between actions, we were interested in how dynamic role assignment, and the implications that follow from roles, can be used to reframe similar action sequences to have very different meanings.

9.2.1 The Authoring Process

For this paper, we look at the third of the three use-cases for story generation. The general setup of the system is as follows in Figure 9-1. At the top level, there are three parts of the ROBOT DETECTIVE domain. First, the authored story is broken down into five images, each representing some significant vignette. Author specifications are then taken from these images and combined with story rulesets. The presentation system produces English sentences in accordance with the generated story variation. RoleModel is the logic system, while ROBOT DETECTIVE is the story domain.

Image 1	2 characters are brothers
Image 2	Introduce 3rd character
Image 3	One brother is romantically involved with the 3rd character
Image 4	The other brother is sad
Image 5	The other brother kills the first

Table 9-1. The describing the fixed image assets of ROBOT DETECTIVE.

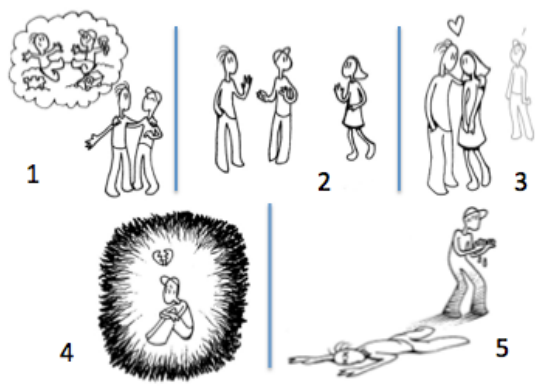

Figure 9-1. The fixed image assets described in Table 9-1.

Each image represents the constituent events of the overall story experience for ROBOT DETECTIVE. In the following section, we will detail the system design of RoleModel and ROBOT DETECTIVE.

9.3 System Design

The overall logic aspect of the system is diagrammed in Figure 9-2 below. In addition to the logic system, we will also describe the refinement process in taking the solved-for logical models and transforming them into a human-readable narrative.

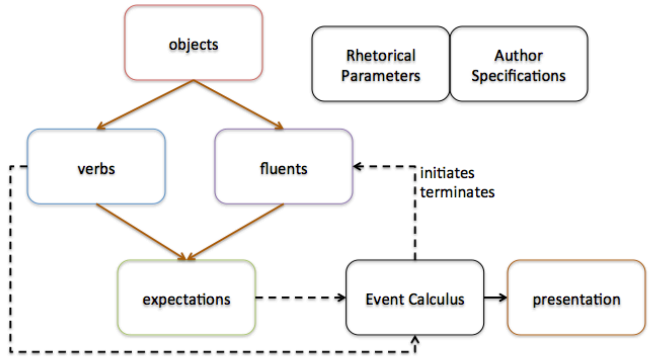

Figure 9-2. A representation of ROBOT DETECTIVE as distinct rulesets.

Due to the stageless nature of answer-set (logic) programming, rulesets are not entirely distinct. Figure 9-2 shows a generalized overview of the system's rulesets and how they inform one another. Objects, by themselves, do not carry meaning; they must be associated with verbs and fluents. Verbs and fluents, although they have some meaning, are used to compose high-level narrative understanding, or expectations. The dotted lines indicate that verbs and expectations do not happen without being placed in time (through the event calculus), while the dotted arrow that points away from the event calculus shows that it functions by initiating and terminating fluents. Such initializations and terminations are dictated through the placement of verbs and expectations in time. Information held together by the event calculus is used to compose the final narrative presentation. Table 9-2, shown below, defines these rulesets.

Objects	Physical objects
Verbs	Atomic actions
Fluents	The state of the world per time-step

Expectations	Story skeletons, patterns of atomic actions
Event Calculus	Manages the initiation and termination of **fluents**, caused by **verbs** and **expectations**
Presentation	Manages the content to reveal to the audience
Parameters	Controls the type of story variation desired
Specifications	Definitions of characters and constituent events

Table 9-2. Overview of the major components of ROBOT DETECTIVE.

The next sub-sections will describe the how the system handles objects, verbs, fluents, expectations, and rhetorical parameters. It concludes with an explanation of the presentation and natural language handling.

9.3.1 Objects

In RoleModel, existents, actions, and their properties are represented by predicates and constants[12] in ASP. ROBOT DETECTIVE has eight potential objects in four object classes: money, weapon, projectile, and gift. The story generator prefers such representations to have a low-level aspect, the object itself, and a high-level meaning, the object's type. Similar hierarchies are necessary and reoccurring in the system design for ROBOT DETECTIVE. The diagram below illustrates how physical objects are represented.

[12] RoleModel uses First-Order Logic (FOL) for its story formalizations. It makes use of predicates to indicate relations and distinguish among sets and sets of sets. For RoleModel, constant values represent existents in the storyworld, which can be bound to variables contained within predicates.

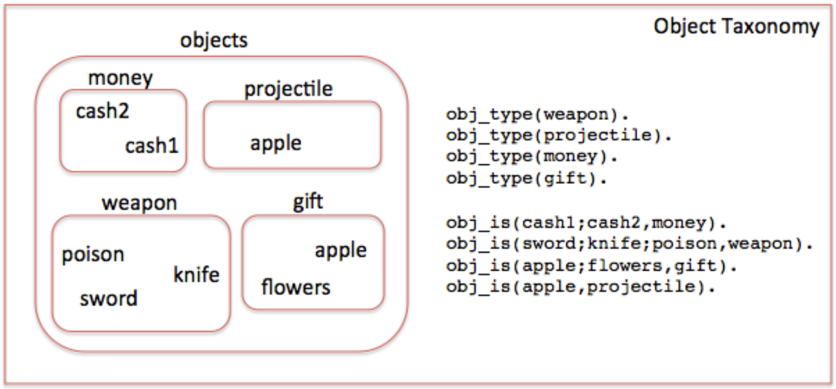

Figure 9-3. Physical objects.

Hierarchies and Sewing Machines

As a result of being stageless, the designer is essentially building a bridge from both ends and the same time. One end is the concrete, and the other, its abstraction. For example, it is necessary for RoleModel to understand how to detect the act of gift-giving in the story world. The physical object that constitutes the gift is unimportant to RoleModel; however, for the presentation of ROBOT DETECTIVE, it is useful to know the particular object. To illustrate: a sewing machine does not care about the color of thread, but the person who wears the shirt most likely does. In other words, both the specific object and its abstraction are important to preserve. For RoleModel, it may not matter whether the gift is an apple or a flower, but, for ROBOT DETECTIVE, this information is important for audience understanding.

On the other hand, objects are connected by relations and events over time; therefore, their uniqueness, rather than their constitution, is important to RoleModel. This can been seen in Figure 9-3 above where we have cash1 and cash2 as objects of type money. Such use of hierarchies reoccurs throughout RoleModels rulesets.

9.3.2 Verbs

In RoleModel, verbs are borrowed from conceptual dependencies. A verb becomes an event once it is assigned a subject and an object, indicated by the P1 and P2 variables in Table 9-3 below. These events do not happen unless they have been assigned a time-step. An event is legal if the appropriate properties of the world are in place. In general, an event at time T will be legal if some set of fluents, at time T-1, permit it to be. If the event happens, then it generally changes some set of fluents from T+1 onward.

EVENT (P1,**verb**,P2).	**Meaning**
verb(**ptrans**(O)).	P1 loses ownership of object O. P2 gains ownership of O. O must not be a gift that was given to P1.
verb(**propel**(O)).	P1 owns object O. P1 throws object O at P2. O must be a projectile. P1 loses object O.
verb(**attacks**(O)).	P1 attacks P2 with object O. O must be a weapon. P1 must own O.
verb(**kills**).	P1 and P2 must be alive. P1 must have attacked P2 at time T-1. P2 is not longer alive from time T+1 and onward.
verb(**mtrans**(M)).	P1 convinces P2 of an expectation. P2 wants this expectation from time T onwards.
verb(**idle**).	P1 does nothing at time T.

Table 9-3. Verbs.

9.3.3 Fluents

Since events do not indicate time of occurrence, the event-calculus can assign verbs to characters, forming an event, and place it within the timeline of the story. The management of these assignments helps to satisfy story and authorial constraints. As these events are placed in time, their associated fluents accommodate the rules and constraints of RoleModel. All fluents represent attributes for a specific character, and they do not become fluents unless they have been assigned a character subject and a time-point. There are 4 categories of fluents in ROBOT DETECTIVE: (1) relationship with another character, (2) physical and emotional state, (3) desires, and (4) inventory of possessions.

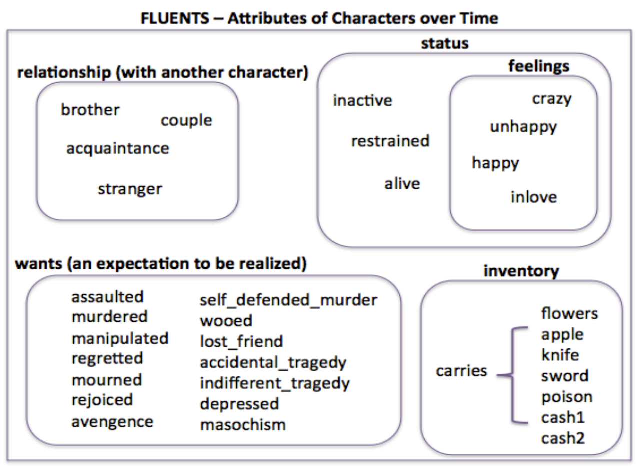

Figure 9-4. Fluents.

Fluent Type	Attribute	Meaning
relationship	is_related(Re,P).	Subject is of Re relation to P
status	attribute(A).	It holds that subject is attribute A
desire	wants(M).	Subject desires expectation
inventory	carries(O).	Subject carries object O

Table 9-4. Fluent types.

Every fluent holds true until terminated and false until initiated (this is the commonsense law of inertia). Fluents are terminated and initiated by the verbs described in the previous sub-section. The logical representation of a fluent reads as follows: holds(attrib(Subject,Attribute),T). This translates to: for the Subject, it holds at time T that this Attribute is true for the Subject.

9.3.4 Story Skeletons

The sewing machine analogy for objects and object types, given in section 4.1.1, carries over for events and story skeletons. Recall that an event is a verb with a subject and an object. A skeleton occurs when some expectation is assigned a subject, object, and time-step. The expectations are listed in the left column of the Table 9-5 below.

Story Skeleton (X,P1,P2,T).	**Meaning**
murdered	P1 kills P2 at time T.
manipulated	P1 convinces P2 to perform an action prior to time T. P2 performs the act at time T. The act is harmful.
regretted	P1 performs a harmful act against P2 prior to time T. At time T, P1 initiates being regretful.
mourned	Someone kills P2 at time T-1. P1 initiates mourning of P2 at time T.
rejoiced	P1 does an act towards P2 at time T-1. P1 initiates rejoicing of P2 at time T.
avengence	P2 kills an acquaintance of P1 prior to time T. P1 performs harmful act against P2 at time T.
self_defended_murder	P2 assaults P1 at time T-1. P1 kills P2 at time T.
wooed	P1 gives an object to P2 at time T-1. The object is a gift.
lost_friend	P2 mourns a death at time T+1. P1 kills someone at T. The person killed is not a stranger to P2.
accidental_tragedy	P1 mourns the loss of P2 and immediately regrets causing the loss of P2. P1 initiates realization of accident.
indifferent_tragedy	P1 wanted to be harmed. P2 causes harm to P1. P1 initiates indifference.
depressed	If at any point, P1 wants to be murdered, then P1 is depressed.
masochism	P1 is depressed at some point before time T. P2 causes harm to P1. (see condition for depressed)
assaulted	P1 assaults P2 by attacking with an object. This object is a weapon.

Table 9-5. Story Skeletons and their meaning.

Filtering out the Nonsense

Schank's story-fitting metaphor carries over and inspires the design of ROBOT DETECTIVE. The meaning of an event is relative to its circumstances—for example, the surrounding events. The skeleton is, therefore, a collection and ordering of events and fluents, purposed for audience interpretation. In ROBOT DETECTIVE, authorial specifications can require and deny story skeletons (i.e., they contain skeleton constraints). In other words, these story skeletons are useful higher-level controls afforded to the author and provided by RoleModel. The author may require a murder to occur, as ROBOT DETECTIVE does, without having to specify the murder weapon or time. That information is, however, assigned within RoleModel,.

In ROBOT DETECTIVE, using skeletons filters out nonsensical stories on two levels: (1) no event can happen on its own, because by themselves, events are meaningless, and (2) the presentation rulesets only deliver relevant patterns of events, relying on story skeletons rather than events for meaning. Many of the skeletons require the character to hold its expectation before enacting it. We represent such desires as fluents that are initiated by the mtrans verb.

Former approaches to generative storytelling utilized character modeling, author modeling, story modeling and reader modeling to create and maintain meaningful variations. The use of pattern detection, as explained through story skeletons, combines these conventions together, controllable through declarative constraints. These models can filter out or prevent nonsensical narrative behavior of which RoleModel declaratively employs.

Types of Story Skeletons

Such patterns of events also overlap, creating unspecified expectations. The solver detects all possible patterns of events in a particular story, which enables RoleModel to refuse unwanted emergent story skeletons. For example, if we aim to vilify a character in our story, we may want to block the occurrence of plausible self-defense. We may either prevent the story from creating that scenario or leave it out of the presentation. This is especially useful for satisfying the parameters for ROBOT DETECTIVE's space of variations, which we talk about in the next sub-section.

We've established that story skeletons implicitly capture patterns of events. RoleModel creates an additional level of abstraction that pivots on whether the story skeleton creates an expectation that is harmful to another character. For the parameters of role that RoleModel is concerned with, harm is the main property of focus. For this purpose, RoleModel only sees such high-level actions in a harmful/unharmful dichotomy.

9.3.5 Rhetorical Parameters

As described in the last section, we parameterize RoleModel's storytelling around role assignments of victim and aggressor. These roles are resultants of harmful events. Harmfulness is categorized on the level of story skeletons, since events, by themselves, do not carry narrative meaning. Table 9-6 below gives the formal definition from RoleModel.

Roles	Condition
role_satisfied(**victim**,P).	P is a victim if P is on the receiving end of a harmful act.
role_satisfied(**aggressor**,P).	P is an aggressor if P causes harm to another.
role_satisfied(**bystander**,P).	P is an bystander if P is neither victim nor aggressor

Table 9-6. Rhetorical parameters, represented by role assignments, and their meaning.

Rhetorical parameters guide the types of variations desired. In RoleModel, the parameters of interest are based off of character role assignments. For the system, each character can be a victim, aggressor, victim-aggressor, or a bystander. In the original RoleModel system, these role assignments were maintained by enabling harm to happen or to nullify its impacts. The logical models behind the justification and nullification of rules was inspired by Abelson's rationalization mechanisms described in our related works section, and were furthered by Schank's story-fitting phenomenon. Schank uses a widely understood narrative around divorce to demonstrate what we described earlier as the Rashomon Effect, where the audience is to reconcile equally plausible, yet contradictory variations of the same story.

> When people divide to tell about their divorce they are telling about the situation the way they understand it and the way they hope their hearers will understand it. They are thus forced, in some sense, to make their story acceptable and easily comprehensible both by their initial attempts to understand the events themselves and by their prior attempts to tell others their story. To achieve this goal, they choose a standard story to tell– a story skeleton– and force the facts to fit the skeleton.

Schank concludes that, "If parts of the story do not fit the skeleton, they ignore them. If the story also fits a skeleton that is not favorable to the tellers, they acknowledge the superficial parallels and then dispute the accuracy of that skeleton in their case."

In Table 9-7 below, we define the nullifiers for victim and aggressor that RoleModel uses for ROBOT DETECTIVE.

Victim Nullifiers	**Aggressor Nullifiers**
if you manipulate someone into doing something to yourself	You were manipulated by another character
if you wanted it to happen	The person you harmed does not care that they were harmed
someone harms you in self defense, because you attacked them	if you cause harm in self defense
if you caused harm to someone in the past and they respond with a later harmful act	if your causing of harm was an accidental byproduct
if you do not care that your friend is dead	if the your friend was murdered and you attempt to avenge the death

Table 9-7. Role Nullifiers inspired by Abelson's rationalization mechanisms and Schank's story skeletons.

9.3.6 Presentation

In addition to building a successful story, RoleModel must satisfy the rhetorical parameters such that any parameterized variation clearly presents its authorial intent to the audience. A combination of factors play into how well this is communicated through the system's output. Four levels of the design process influence the final presentation: (1) the initial design and author specifications for the story, (2) the quality of story being generated, (3) the ruleset that sews together the narrative for the audience, and (4) the translation of logical statements into a digestible story experience. (1) and (4) are handled outside of RoleModel. In the previous sub-sections, we described how we arrive at (1) and (2). In this sub-section, we describe the design process for (3) and (4).

In the diagram below, the third level (3) is outlined by the box labeled "presentations." The box labeled Natural Language Generation indicates the fourth level (4).

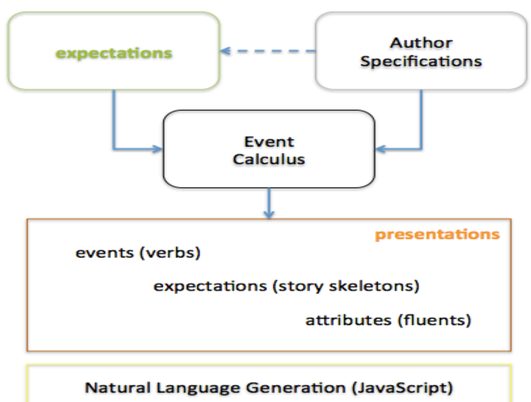

Figure 9-5. From logical output to presentation.

The presentations box is part of the logical rulesets of RoleModel, while the bottom-most box exists after the story has been solved for. In the case of ROBOT DETECTIVE, the final, outer-most layer of the presentation is created through a JavaScript program that takes the story model generated by RoleModel as input.

It seemed that Al picked a fight with Parry. Eliza had manipulated Al. Parry murdered Al. He killed in self defense against Al. Al attacks Parry with a sword. Parry attacks Al with a knife. He kills Al.

```
presentation(motiv(assaulted,al,parry), 5 );
presentation(motiv(manipulated,eliza,al), 5 );
presentation(motiv(murdered,parry,al), 5 );
presentation(motiv(self_defended_murder,parry,al,5), 5 );
presentation(svo(al,attacks(sword),parry), 5 );
presentation(svo(parry,attacks(knife),al), 5 );
presentation(svo(parry,kills,al), 5 );
```

Figure 9-6. Story output with image and natural language interpretation

Each presentation predicate indicates a sentence of the narration and the time-step it occurs. For ROBOT DETECTIVE, time-steps 1, 2, 3, 4, and 5 correspond to the five images described in Section 3. Time-step 0 is a preface and anything beyond image 5 are simply closing remarks. Allowing RoleModel these bookend time-steps gives it room to generate a slight amount of elaboration around the pre-authored story. Below is an example story generated for the ROBOT DETECTIVE JavaScript presenter.

```
presentation(expect(assaulted,al,parry), 0 );
presentation(attrib(al,rel(brother,parry)), 1 );
presentation(attrib(eliza,rel(acquaintance,al)), 2 );
presentation(attrib(eliza,rel(acquaintance,parry)), 2 );
presentation(event(al,ptrans(apple),eliza), 2 );
presentation(event(eliza,ptrans(flowers),al), 2 );
presentation(attrib(eliza,rel(couple,al)), 3 );
presentation(event(eliza,mtrans(expect(assaulted,al,parry)),al), 3 );
presentation(attrib(parry,unhappy), 4 );
presentation(expect(assaulted,parry,al), 4 );
presentation(expect(assaulted,al,parry), 5 );
presentation(expect(manipulated,eliza,al), 5 );
presentation(expect(murdered,parry,al), 5 );
presentation(expect(self_defended_murder,parry,al,5), 5 );
presentation(event(al,attacks(sword),parry), 5 );
presentation(event(parry,attacks(knife),al), 5 );
presentation(event(parry,kills,al), 5 );
presentation(expect(rejoiced,eliza,al), 6 );
```

Figure 9-7. Example story output.

Shown in the example output above, RoleModel produces three types of sentences: expectations, attributes, and events. Expectations are the high-level story skeletons, which carry narrative meaning. The selected attributes are notable properties of characters relevant to the narrative at a given time. Although events do not hold narrative meaning, they are often presented in conjunction with other events to convey higher-order expectations dictated by the story skeletons. The image above shows a captures the output, image, and caption of time-step 5. The predicates and its parameters are associated with function names and objects directly in the JavaScript program, set up for conjugation and pronouns.

Story-Gen Paradigms	Declarative-Rhetorical Paradigm
World-Model	Character Motivation Rulesets
Audience-Model	Presentation Rulesets
Author-Model	Story Skeletons and Rhetorical Params
Story-Model	Fixed Assets and Story Specifications

Table 9-8. Generative storytelling paradigms listed on the left with RoleModel's approach on the right

Chapter 10 – RoleModel Results

In practice, RoleModel is one approach to the Declarative-Rhetorical Model for storytelling. Instead of focusing on character models or story models, the system uses constraint-satisfaction to operationalize all of the above. Therefore, constituent story events are declarative constraints on a story world, maintaining the integrity of an artist's specifications.

RoleModel generates supplementary variations for ROBOT DETECTIVE. The operationalization of these variations is represented through models of traditional authorial conventions or idioms, the motivations of characters, the perception of the audience, the desired structure for story, etc. Rhetorical parameters are developed to direct the space of variations. The quality of the story is established by (1) the artistic constraints of the author, and (2) rulesets that operationalize the supplementary outcomes.

In our study, We used 48 people (25 in Trial A and 25 in Trial B) to test **RoleModel/ROBOT DETECTIVE**. Each trial used the same image assets with two sets of generated captions. Eliza was the main character in the stories for both trials. The stories were generated to have Eliza change roles from aggressor to victim. We wanted to test whether participants would see the change in the two different variations, even though the same illustrations were used. Stories A and B were randomized, so the participants viewed them in various orders.

Trial A	Once upon a time...Al picked a fight with Parry. It was the case that Al and Parry are brothers. One day, Eliza and Al became acquaintances. She and Parry became acquaintances. Al gave apples to Eliza. Eliza gave flowers to Al. Then, Eliza and Al fell in love and became a couple. Al had won the affections of Eliza. Eliza had won the affections of Al. She convinced Al to pick a fight with Parry. For whatever reason, Parry was unhappy. He picked a fight with Al. It seemed that Al picked a fight with Parry. Eliza had manipulated Al. Parry killed Al. He killed in self defense against Al. As a result, Eliza was glad about what happened to Al.
Trial B	It was the case that Al and Parry are brothers. One day, Eliza and Al became acquaintances. She and Parry became acquaintances. She gave apples to Al. Then, Eliza and Al fell in love and became a couple. Parry picked a fight with Eliza. Eliza had won the affections of Al. For whatever reason, Parry was unhappy. He picked a fight with Al. It seemed that Parry killed Al. He didn't intend to harm Al. As a result, Eliza grieved the loss of Al. Parry grieved the loss of Al. In the end, Parry gave flowers to Eliza. Parry had won the affections of Eliza.

Table 10-1. The image captions for Trial A and Trial B.

	Observed Data		Simulated Truth		Two Sample Binomial Test
	Trial A, n=25	Trial B, n=23	Trial A	Trial B	p-value
Eliza-agg	25	0	Yes	No	4.262e-12
Eliza-victim	1	16	No	Yes	2.087e-06

Table 10-2. Results of the ROBOT DETECTIVE study.

In Story A, the story generator set Eliza as the aggressor, and all 25 participants correctly identified this. In Story B, Eliza was not set as the aggressor, and all of the 23 participants correctly identified this. Participants were also asked if Eliza was a victim in Story A and B. The specification for Story B contained the constraint required_role(victim,eliza) while Story A did not. One of twenty-five and sixteen of twenty-three respectively said she was. The different perception of participants between Story A and B is verified with a two-sample binomial test with p-value of 2.087E-6. Overall, the participants demonstrated the story generator's successful portrayal of Eliza's different roles in the two stories.

Parry and Al were set to be unaggressive in both plots. Participants had trouble distinguishing whether or not Parry or Al were aggressors, but this was not the focus of the study. Neither Parry nor Al were set as victims in the story generator; therefore, the generator decided. The victimhood of these players was not of interest.

10.1 DISCUSSION

In our discussion section, we explain our Declarative-Rhetorical Model around the ROBOT DETECTIVE system. We identify important takeaways and benefits of using this approach. As discussed in our Related Works section, storytelling AI aims for high quality, high variations, and high flexibility, with low authorial effort. A major challenge for generative storytelling is creating a large space of variations that are meaningful, filtering out narrative nonsense. When designing ROBOT DETECTIVE, we worked towards controlled story variations through declarative representation of narrative models.

First, our declarative approach makes use of models taken from story-generation paradigms. Table 9-8 above places generation models with the equivalent rulesets and formalizations of RoleModel. To ensure meaningful stories, we balance over/undergeneration by the means mentioned in the Declarative-Rhetorical column in Table 9-8. Overgeneration results in lots of variety, but it also allows some nonsensical stories to be generated. With undergeneration, all the stories make sense, but many interesting stories are rejected. RoleModel provides this balance by leveraging rhetorical parameters.

Second, this approach is designed to optimize authorial leverage by providing the following:

- Authorial Effort: Viewing the constituent events as author constraints using traditional authoring as the cornerstone for a story world; this is easy to do after the establishment of the basic rulesets.
- Authorial Flexibility: Using low-cost assets to target and operationalize specific types of variations, whether discourse or ideological variations; these are easy to change and reuse.

For ROBOT DETECTIVE, the authorial practice is as follows:

- Establish the who, what, and where of a story in the system, such that they are operationally meaningful (constituent events)

- Designate a scenario of what has actually occurred (optional constituent events)
- Decide on aspects of that are locked in through artifacts and witness accounts (constituent events)
- Parameterize the story around designating guilt and innocence (rhetorical parameters)
- Operationalize properties that exaggerate, enhance, and nullify parameter settings (supplementary variations)
- Develop a means for interactions and presentation (supplementary variations)

In this chapter, we discussed the influences on the RoleModel system, in particular, the Rashomon Effect. We then looked at the design and implementation of the ROBOT DETECTIVE story space, described the study, and laid out the results of generating narrative parameterized by role assignments (of victim or aggressor). We were interested in whether managing supplementary details would yield meaningful variations. The results show that a single character can be seen as both victim and aggressor, depending on the narration of the events.

Chapter 11 – Conclusion

> "We can only see a short distance ahead, but we can see plenty there that needs to be done." (Turing, 1950).

This dissertation explores how we can understand and improve storytelling on computers. The three major contributions of this work are: (1) authorial leverage, (2) supplementary/constituent dichotomy in storytelling, (3) RoleModel. In Chapter 2 we discuss the overall problem overview, and in Chapters 3, 4, and 5, we describe the three major contributions of this work.

Chapter 2-5 dives into the history of the theory of AI and storytelling. It describes what we mean by telling stories with computers, and what the challenges are. It moves the discussion away from the content/discourse dichotomy, which caters to the user, toward a constituent/supplementary dichotomy, that caters to the author's leverage.

Chapter 6-7 introduces the concept of Authorial Leverage. This chapter also provides a general evaluation of various AI storytelling systems, with a specific emphasis on the Drama Management system.

Chapter 8-10 presents RoleModel, which is built around Robot Detective, a story with fixed art assets and generates variations around role parameters. Even with the fixed art assets, text changes enabled users in the study to identify different roles for the main characters. It describes a system that implements and evaluates a system built under the constituent/supplementary principle.

While other studies have attempted to address the problem of having computers generate better stories, these approaches often come at the cost of increasing authorial burden in other ways. RoleModel, however, provides a way to create meaningful variations in story by leveraging human interpretation. This is an early example of a more author-centric design of an AI storytelling system. By making use of the constituent/supplementary paradigm, it grants us better AL.

11.1 Future work

11.1.1 Game and Narrative Design

The game industry incurs a huge cost when producing art and story assets. This author centric approach reduces the amount of expensive story artifacts by leveraging meaningful supplementary variations, using low-cost variations that are simpler for a computer to manage.

11.1.2 Game Studies

This new theory of story design provides a more author centric model for analysis and creation. The traditional view of story relies on a content/discourse dichotomy. Future work focused on the supplementary/constituent dichotomy and Authorial Leverage would be necessary for deeper understanding of story design.

11.1.3 Intelligent Narrative Technologies

Areas of refinement such as Natural Language Processing still have many possible places to improve, in order to provide acceptable presentation of stories. Often with AI systems, the generated language can confuse and take away from the user's experience. Also, future forms of variations have yet to be explored.

Chapter 12 – REFERENCES

Abelson, R. P. (1963). Computer Simulation of Personality, Frontier of Psychological Theory: Computer Simulation of "Hot" Cognition. Yale University. John Wiley & Sons Inc.

Abbott, H.P. (2002). The Cambridge Introduction to Narrative. Cambridge University Press.

Bailey, P. (1999). Searching for storiness: Story-generation from a reader's perspective. In Working notes of the Narrative Intelligence Symposium (pp. 157-164).

Bartlett, F. (1932). Remembering. Cambridge University Press, Cambridge.

Carbonell Jr, J. G. (1978). POLITICS: Automated ideological reasoning. Cognitive Science, 2(1), 27-51.

Cavazza, M., & Pizzi, D. (2006). Narratology for interactive storytelling: A critical introduction. In Technologies for Interactive Digital Storytelling and Entertainment (pp. 72-83). Springer Berlin Heidelberg.

Chatman, S. (1978). Story & Discourse. Cornell University Press.

Chen, S., Nelson, M. J., Sullivan, A., & Mateas, M. (2009). Evaluating the Authorial Leverage of Drama Management. In AAAI Spring Symposium: Intelligent Narrative Technologies II (pp. 20-23).

Chen, S., Smith, A. M., Jhala, A., Wardrip-Fruin, N., & Mateas, M. (2010). Rolemodel: towards a formal model of dramatic roles for story generation. In Proceedings of the Intelligent Narrative Technologies III Workshop (p. 17). ACM.

Crawford, C. (2012). Chris Crawford on interactive storytelling. New Riders.

Frasca, G. (2003). Frasca response to Mateas. In N. Wardrip-Fruin and P. Harrigan. (Eds.), First Person: New Media as Story, Performance and Game. Cambridge MA:The MIT Press.

Gebser, M., Kaminski, R., Kaufmann, B., & Schaub, T. (2012). Answer Set Solving in Practice. Synthesis Lectures on Artificial Intelligence and Machine Learning, 6(3), 1-238.

Gennette, G. (1979). Narrative Discourse, An Essay in Method. Cornell University Press.

Kurosawa, A. (1950). Rashomon.

Ryan, M. L. Avatars of story. U of Minnesota Press, 2006.

Laurel, B. (2013). Computers as theatre, Second Edition.

Mateas, M. (2000). A neo-aristotelian theory of interactive drama. In Working notes of the AI and Interactive Entertainment Symposium.

Mateas, M. (2001). A preliminary poetics for interactive drama and games. Digital Creativity, 12(3), 140-152.

Mateas, M. (2002). Interactive drama, art and artificial intelligence.

Mateas, M., Domike, S., & Vanouse, P. (1999). Terminal time: An ideologically-biased history machine. AISB Quarterly, Special Issue on Creativity in the Arts and Sciences, 102, 36-43.

Mateas, M., & Sengers, P. (Eds.). (2003). Narrative intelligence (Vol. 46). John Benjamins Publishing.

Mateas, M., & Stern, A. (2005). Structuring Content in the Façade Interactive Drama Architecture. In AIIDE (pp. 93-98).

Mateas, M., & Sengers, P. (Eds.). (2003). Narrative intelligence (Vol. 46). John Benjamins Publishing.

McCoy, J., Treanor, M., Samuel, B., Mateas, M., & Wardrip-Fruin, N. (2011). Prom Week: social physics as gameplay. In Proceedings of the 6th International Conference on Foundations of Digital Games (pp. 319-321). ACM.

McKee, R. (1997). Story: Substance, Structure, Style, and the Principles of Screenwriting. HarperCollins.

Montfort, N. (2009). Curveship: an interactive fiction system for interactive narrating. In Proceedings of the Workshop on Computational Approaches to Linguistic Creativity (pp. 55-62). Association for Computational Linguistics.

Mueller, E. T. (2010). Commonsense reasoning. Morgan Kaufmann.

Murray, J. H. (1997). Hamlet on the holodeck: The future of narrative in cyberspace. Simon and Schuster.

Nelson, M. J., Mateas, M., Roberts, D. L., & Isbell, C. L. (2006). Declarative optimization-based drama management in interactive fiction. Computer Graphics and Applications, IEEE, 26(3), 32-41.

Ontañón, S. and Zhu, J. (2010). Story and Text Generation through Computational Analogy in the Riu System. AIIDE.

Orkin, J., & Roy, D. (2007). The restaurant game: Learning social behavior and language from thousands of players online. Journal of Game Development, 3(1), 39-60.

Pfister, M. (1977). The Theory and Analysis of Drama. Cambridge University Press. London and Germany.

Riedl, M. & Young, R. M. (2004). An Intent Driven Planner for Multi-Agent Story Generation. Autonomous Agents and Multi-Agent Systems.

Riedl, M. O., Rowe, J. P., & Elson, D. K. (2008). Toward intelligent support of authoring machinima media content: story and visualization. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Roberts, D. L. & Isbell, C. L. (2008). A survey and qualitative analysis of recent advances in drama management. International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning, 4(2), 61-75.

Russell, S., Norvig, P., & Intelligence, A. (1995). A modern approach. Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs, 25.

Ryan, M. (2001). Beyond Myth and Metaphor. http://gamestudies.org/0101/ryan/.

Schank, R. C. (1990). Tell me a story: A new look at real and artificial memory. Charles Scribner's Sons.

Skorupski, J. & Mateas, M. (2010). Novice-friendly Authoring of Plan-based Interactive Storyboards, Artificial Intelligence and Interactive Digital Entertainment.

Smith, A. M., & Mateas, M. (2010, August). Variations forever: Flexibly generating rulesets from a sculptable design space of mini-games. In Computational Intelligence and Games (CIG), 2010 IEEE Symposium on (pp. 273-280). IEEE.

Smith, A. M., & Mateas, M. (2011). Answer set programming for procedural content generation: A design space approach. Computational Intelligence and AI in Games, IEEE Transactions on, 3(3), 187-200.

Sullivan, A., Chen, S., & Mateas, M. (2008). Integrating Drama Management into an Adventure Game. In AIIDE.

Sullivan, A., Chen, S., & Mateas, M. (2009) From Abstraction to Reality: Integrating Drama Management into a Playable Game Experience, AAAI 2009 Spring Symposium on Interactive Narrative Technologies II: AAAI Press.

Turing, A. M. (1950). Computing machinery and intelligence. Mind, 433-460.

Turner, S. R. (1993). Minstrel: a computer model of creativity and storytelling.

Thue, D., Bulitko, V., & Hamilton, H. J. (2013). Implementation Cost and Efficiency for AI Experience Managers. In Ninth Artificial Intelligence and Interactive Digital Entertainment Conference.

Weyhrauch, P. & Joseph B. Guiding interactive drama. Pittsburgh, PA: Carnegie Mellon University, 1997.

y Perez, R.P. & Sharples, M. (2004). Three computer-based models of storytelling: BRUTUS, MINSTREL and MEXICA. Knowledge-Based Systems, Volume 11.

http://www.theatlantic.com/technology/archive/2014/06/when-parry-met-eliza-a-ridiculous-chatbot-conversation-from-1972/372428/