A COMPUTATIONAL MODEL OF SUSPENSE FOR THE AUGMENTATION OF INTELLIGENT STORY GENERATION

A Thesis
Presented to
The Academic Faculty

by

Brian O'Neill

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology
December 2013

A COMPUTATIONAL MODEL OF SUSPENSE FOR THE AUGMENTATION OF INTELLIGENT STORY GENERATION

Approved by:

Dr. Mark Riedl, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Ashok Goel
School of Interactive Computing
Georgia Institute of Technology

Dr. Ashwin Ram
School of Interactive Computing
Georgia Institute of Technology

Dr. Brian Magerko
School of Literature, Media, and Communication
Georgia Institute of Technology

Dr. Pablo Gervás
Departamento de Ingeniería de Software e Inteligencia Artificial, Facultad de Informática
Universidad Complutense de Madrid

Date Approved: 15 November 2013

To Heather,

and to my family,

I love you all.

ACKNOWLEDGEMENTS

There are an incredible number of people who deserve my thanks, and without whom I doubt I could have completed this work. For their encouragement, their support, and their guidance, each of these people have my immense gratitude, in ways that I can only attempt to convey here.

- My parents, Patrick and Jill O'Neill, and my brother, Timothy, for never doubting for a moment that I would succeed at anything I put my mind to, and for their omnipresent faith, support, love, and reassurances when I doubted whether I could overcome the obstacles in my way.

- Mark Riedl, my advisor, for being my mentor for the last five years and several months, for his consistent support and guidance of my research, and for reading the majority of this document while simultaneously caring for his newborn son.

- My committee, Pablo Gervás, Ashok Goel, Brian Magerko, and Ashwin Ram, for their time, their effort, and for the opportunities to work with them that they gave to me.

- Two past advisors, Steve Cooper and Mark Guzdial, for starting me down the path of research at Saint Joseph's University and Georgia Tech, respectively, and for freely giving me advice even after our formal student-advisor relationships had ended.

- Everyone in the Entertainment Intelligence Lab, notably Boyang Li, Alex Zook, Hong Yu, and Stephen Lee-Urban, each of whom has helped me shape this work and put it into words.

- Kimberly Xu, who got lost on campus with me during our first visit, leading to a friendship that would last through our time at Georgia Tech and beyond. Thank you for always being available for walks, venting sessions, and beer.

- For being mentors, voices of sanity, and, of course, friends: Brian Dorn, Allison Tew, David Roberts, Hank Carter, Chad Stolper, Catherine Grevet, and Briana Morrison.

- Heidi Ellis, Dean Saeed Ghahramani, and Western New England University, for hiring me before I finished my degree, and kindly giving me the time to finish this work without distraction, so that I could be a better researcher and a better educator.

- And finally, Heather Borgoyn, for her unwavering love, support, and confidence in me, especially when my own doubts and fears reached their peak, and in spite of our distance and time apart. I love you, and I am so glad to have that time and that distance end.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Narrative as entertainment plays a central role in many forms of entertainment media, including novels, movies, games, and theatre. One of the reasons for the prevalence of storytelling in human culture may be due to the way in which narrative is used as a cognitive tool for situated understanding. Expert storytellers who craft narratives for entertainment structure their narratives to be aesthetically pleasing to the audience. Computer scientists have tried for more than three decades to determine whether, and how, intelligent computational systems can create aesthetically pleasing narratives from scratch. One of the many tools that expert storytellers use to make stories aesthetically pleasing is suspense.

In this dissertation, I present DRAMATIS, a computational human behavior model of suspense based on Gerrig and Bernardo's definition of suspense. In this model, readers traverse a search space on behalf of the protagonist, searching for an escape from some oncoming negative outcome. As the quality or quantity of escapes available to the protagonist decreases, the level of suspense felt by the audience increases. The major components of Dramatis are a model of reader salience, used to determine what elements of the story are foregrounded in the reader's mind, and an algorithm for determining the escape plan that a reader would perceive to be the most likely to succeed for the protagonist. I evaluate my model by comparing its ratings of suspense to the self-reported suspense ratings of human readers. Additionally, I demonstrate that the components of the suspense model are sufficient to produce these human-comparable ratings.

Additionally, I present an approach for knowledge engineering based on qualitative methods. Dramatis is a knowledge-intensive system, requiring representations

of actions in the world and readers' genre knowledge. This qualitative knowledge engineering methodology allows for the conversion of a natural language corpus into a collection of knowledge structures in a way that mitigates engineer bias. Knowledge engineers annotate the corpus using an iterative coding process. These annotations then provide context for the creation of knowledge structures.

CHAPTER I

INTRODUCTION

Narrative as entertainment, in the form of oral, written, or visual storytelling, plays a central role in many forms of entertainment media, including novels, movies, games, and theatre. One of the reasons for the prevalence of storytelling in human culture may be due to the way in which narrative is used as a cognitive tool for situated understanding [14, 29]. This narrative intelligence is central in the cognitive processes that we employ across a range of experiences, from entertainment contexts to active learning.

People create and tell stories on a daily basis because of their power to persuade and entertain. Audiences become immersed in stories, being mentally transported to the narrative world where they forget the rules and expectations of the real world [29]. Imbuing computers with the capacity to create, understand, and tell stories opens an expanse of opportunities. In the realm of games and entertainment, computers would be able to adapt and produce stories faster than human storytellers. One consequence of this speed, without loss of quality, is the potential for stories on-demand. Given a topic or concept, it may one day be possible for an intelligent system to instantaneously create a story that humans find entertaining and exciting. Within games, such speed may allow for never-ending games, as the game world may always be expanded with more characters and quests. In these game contexts, we may also desire characters capable of interacting with human readers or players. By engaging with and responding to the audience, interactive characters and stories provide a richer experience than static characters. Storytelling systems with such interactivity will also require the ability to modify the story in order to appropriately

react to actions taken by human players/audiences, particularly when players take unexpected actions, thereby making the original story no longer viable.

The value of computational storytelling is not limited to entertainment. Humans use stories in educational and training scenarios, as a means of providing context and opportunities for practice. Without these stories, it is difficult for students to ground these lessons in the real-world. For example, training systems can create interactive scenarios where the learner has to apply previous lessons. Knowing the value of student engagement in learning, we want to be certain that the training system is creating realistic and engaging scenarios (a specialized form of story), so that students take away the appropriate knowledge in realistic contexts.

As virtual agents become more common in healthcare [8, 40], there will be a need for agents that both create and understand stories in order to effectively interact with patients. Intelligent agents will need to be able to understand the stories that patients use to explain their symptoms and histories. Additionally, virtual agents will need to be able to create a rapport with patients [8], requiring the capacity to understand their stories and relate to them by sharing their own stories. Understanding stories also provides the potential to adapt and reuse stories in future interactions. However, there is more to stories than being able to create and understand. Stories need to be entertaining in order to be effective in each of these contexts, and in this aspect, computer-generated stories are often lacking.

Expert storytellers who craft narratives for entertainment—films, novels, games, etc.—often structure their narratives to be aesthetically pleasing to the viewer, reader, or player. The idea that story structure is correlated with audience enjoyment dates back to Aristotelian notions of drama [4] as well as more recent narrative theories (e.g. Freytag [27]). Computer scientists have tried for more than three decades to determine whether, and how, intelligent computational systems can create aesthetically pleasing narratives from scratch. Zagalo et al. [78] argue for the use of dramatic

arc and intelligent emotion detection as a means of producing aesthetically pleasing narratives in story generation and storytelling systems. Similarly, Szilas has called for an increased focus on the audience's emotional involvement when designing interactive narrative [71]. However, story generation and interactive narrative systems remain unreliable at creating stories that human audiences find aesthetically pleasing. The systems that do consider aesthetics do so by forcing works to conform to ideal dramatic arcs or models of tension (e.g. [50, 62]), or by evaluating stories according to some author-defined model of goodness (e.g. [55]). In general, story generation and interactive narrative systems focus on aesthetics from the point of view of the author or designer, rather than the reader or player.

While there are a number of strategies for making stories aesthetically pleasing, one common approach is the use of suspense, which has been found to contribute to reader enjoyment [72]. Suspense, and its effect on audiences has been studied extensively by narratologists and psychologists. While scholars have presented varying definitions of suspense, most agree that uncertainty about an undesirable outcome is a key component. In this dissertation, I introduce a formal computational model of the following definition of suspense from Gerrig and Bernardo:

> Readers feel suspense when led to believe that the quantity or quality of paths through the hero's problem space has become diminished. [30]

The definition presented above comes from a psychological perspective of suspense. In Chapter 2, I describe the variety of definitions of suspense in more detail, and defend my selection of the above definition. Formalizing this model compels us to more carefully define the details of the model than is typically necessary in psychology [49]. As a result, I will reformulate Gerrig and Bernardo's model, framing it as a search space which a reader traverses on behalf of the protagonist. Additionally, I define "quality," a term introduced in the above definition, to mean the likelihood of success, as perceived by the reader, of a particular path through that search space.

Additionally, formalizing this model requires defining exactly what forms of suspense my model will be able to understand. In this dissertation, I will describe a number of suspenseful stories, and categorize the sources of suspense, with the intent of defining the limits of my suspense model.

Formalizing the psychological model also requires that we accurately represent human knowledge about stories. Further, this requires that I have strong methods for acquiring that knowledge. I will introduce a method for engineering domains for knowledge-intensive systems, such as my computational model of suspense, that makes use of qualitative research methods.

By developing a computational model of suspense, I support the larger goals of story generation and interactive narrative. This computational model can serve as a heuristic or evaluation metric, leading to stories and interactive narratives that are more suspenseful. Additionally, this model of human aesthetics emphasizes the audience's response to a creative artifact, rather than focusing on an authorial model of aesthetic.

1.1 Research Questions and Contributions

Given the goals of creating models of human aesthetics for computationally creative systems, the existence of psychological models of suspense that can be formalized and represented computationally, and the desire for computational models to be supported by human knowledge, I propose the following research questions:

> **RQ1.** How can a computational system reason about suspense responses to narrative content?
>
> **RQ2.** How can we utilize qualitative methods techniques to support knowledge engineering?

In response to RQ1, I propose Dramatis, a computational model of suspense based on the definition of suspense stated above. Dramatis is a knowledge-intensive

system, requiring information about stories and genre. As a result, and in response to RQ2, I propose a method of knowledge engineering based on qualitative research techniques.

In addition, I pose the following thesis statement:

> A computational model of suspense, utilizing a memory model and goal selection process, produces ratings of relative suspensefulness comparable to those self-reported by human readers of equivalent natural language stories.

The major contributions of this thesis are:

- The formalization of Gerrig and Bernardo's definition and psychological model of suspense.
- A computational human behavior model of suspense, based on psychological and narratological understandings of suspense:
  - which has been validated against human self-reported suspense levels, and
  - the sufficiencies of the components of the model have been demonstrated.
- An approach to knowledge engineering, used to collect knowledge for evaluation purposes, that makes use of validated qualitative methods techniques.

1.2 Reader's Guide

The remainder of this dissertation is organized as follows. Chapter 2 describes related work in story comprehension, planning, suspense, and systems that consider aesthetics as part of their creativity. Additionally, I provide a taxonomy of suspense, in order to clarify the sources of suspense which my system is best capable of modeling. Chapter 3 describes Dramatis, my suspense model, in detail. Chapter 4 describes the set of knowledge engineering procedures, based on crowdsourcing and qualitative methods

techniques, which I used to collect scripts and planning operators for Dramatis and its evaluation. This knowledge engineering procedure thus provides Dramatis with the knowledge necessary for a comparison with human readers. Chapter 5 describes an overall evaluation of Dramatis and comparison to human readers, as well as two ablative tests to determine the value of Dramatis components. Finally, Chapter 6 summarizes the dissertation and briefly describes future work.

CHAPTER II

BACKGROUND

In this chapter, I introduce the concept of narrative, and then shift to a discussion of work related to psychological and narratological understandings of suspense, leading to the particular definition of suspense that I use in this research. I provide background into story comprehension and planning, two fields that inform the Dramatis suspense model. I will also discuss intelligent story generation and interactive narrative, focusing on systems that have made attempts to model aesthetics as part of the generation process. I will finish the chapter by returning to suspense and introducing a small number of categories of suspense, as a means of defining the limits of my suspense model.

2.1 Narrative

Narrative is prevalent throughout human culture. We use narratives not only to entertain, but also to explain. We create and share narratives in order to explain the world around us to others. Prince defines narrative as:

> **Definition 1 (Narrative):** The representation... of one or more real or fictive events communicated by one, two, or several (more or less overt) narrators to one, two, or several (more or less overt) narratees [63].

Prince's definition continues on to emphasize the necessity of events. Simple facts such as "Mary is tall" do not constitute events, let alone narratives. However, "The goldfish died" does describe an event and, though uninteresting, fits this definition of a narrative. The "main incidents" of a narrative form the plot, where plots may be structured. Example story structures include Freytag's triangle [27] and Aristotelian

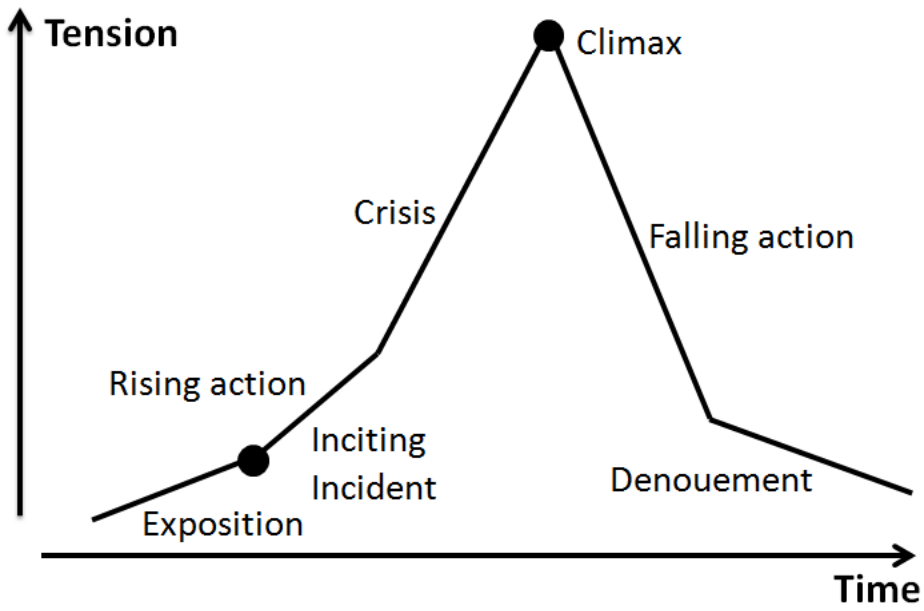

Figure 1: A typical Aristotelian dramatic arc

dramatic arcs [4]. Figure 1 shows a canonical Aristotelian dramatic arc.

Narratologists distinguish between the events of a story and how those events are presented. The *fabula* is the events of the story world [1]. All events are in the fabula, regardless of their order of presentation, if they are presented to the audience at all. In contrast, the *sjuzhet* is an ordering of a subset of the elements of the fabula [1]. It is possible, and common, for a sjuzhet to exclude some elements of the fabula.

Consider the film *Back to the Future*. The majority of the film takes place in 1955, while the beginning and end of the film takes place in 1985. The events of the movie make up the fabula, and the movie itself is one possible sjuzhet. One can imagine another version in which the events of the film are shown in chronological order. This would be a different sjuzhet, but it would still come from the same fabula. A third sjuzhet could be created by including "deleted scenes," as events are now shown that still occurred, but were missing from the other presentations.

2.2 Defining Suspense

Suspense is a common means in storytelling for affecting the audience. However, there is no single consensus definition for suspense, so in this section, I will outline a variety of definitions from psychology, narratology, and entertainment theory, with a focus on four attributes that are common among these definitions: uncertainty, desirability of outcomes, disparity in knowledge between audience and characters, and affinity for the characters. This discussion will ultimately lead to the selection of the particular definition of suspense used in my computational model of suspense.

Many definitions of suspense include the notion of *uncertainty* of an outcome. Abbott describes the key to suspense as the possibility that the events of the story do not turn out according to the expectations of the audience [1]. Similarly, surprise is the feeling that is generated when things *do* turn out differently, thereby violating audience expectation. Taking a narratological view of suspense, Abbott also argues that suspense is a lack of closure in a narrative. Authors appear to satisfy the audience's need for closure, only to revoke this closure before it becomes certain.

In his look at emotion in narrative and film, Tan [72] describes suspense as a "narrative procedure" used to increase the interest of the audience. Tan argues that suspense in narrative or film requires a significant outcome, typically for the protagonist, that is likely, but not certain to occur [72].

Gerrig and Bernardo's definition of suspense also requires an uncertain outcome. According to Gerrig and Bernardo, authors induce suspense by reducing the quantity or quality of plans available to the protagonist of a story for avoiding some negative outcome [30]. Readers act as problem-solvers on behalf of the protagonist, trying to come up with solutions for averting the negative outcome. When readers can only devise low quality plans, or struggle to come up with any plans for a hero to escape the predicament, the perception of suspense increases. Thus, the degree of suspense is correlated with the reader's uncertainty over the means of escape for a

hero. Gerrig and Bernardo studied self-reported suspense levels for several excerpts of stories in order to test this hypothesis. In these studies, they found that readers reported higher suspense levels when story excerpts suggested potential escapes and then quickly eliminated them, thus reducing the quantity of available plans for the protagonist. Similarly, Comisky and Bryant [22] showed that the suspense felt by a film audience is lowest when the fate of the protagonist is absolutely certain, and highest when the protagonist's chances for survival were low.

While Gerrig and Bernardo define suspense from a problem-solving perspective, Ortony, Clore, and Collins [59] consider it in terms of emotion and appraisal theory: From this perspective, suspense involves "a Hope emotion and a Fear emotion coupled with a cognitive state of uncertainty." They further note that uncertainty alone is insufficient for suspense. The uncertain event must also have consequences that are significantly desirable or undesirable to the audience.

In his study of the psychology of suspense in drama, Zillmann notes the emphasis on uncertainty in definitions of suspense, but finds it problematic [80]. His concerns are primarily based in the fact that suspense is often considered a pleasant experience, but uncertainty is not an impetus for pleasant feelings. Zillman does not reject uncertainty as a component of suspense. However, he does believe that uncertainty is overvalued as a factor. Rather, Zillmann focuses on the presentation of desirable and undesirable outcomes as a means of creating suspense.

Like uncertainty, *desirability of outcomes* is common to suspense definitions. Zillmann adopts a definition from Carroll as a guide in his discussion of suspense [80]. Carroll describes suspense as the consequence of a narrative event that has two possible outcomes: one outcome that is morally right and unlikely, the other that is morally incorrect and likely to occur. Thus, suspense is created not only through uncertainty, but through the introduction of the undesirable outcomes.

Note that many of the definitions above cited desirability of outcome in their

definitions. Recall that Tan's definition of suspense required a "significant" outcome with the fate of the protagonist at stake [72]. It is not hard to believe that readers would consider one of the possible outcomes to be more desirable than the other if the protagonist's fate is at risk. Gerrig and Bernardo's definition also required that the protagonist was facing an undesirable outcome [30]. Readers feel suspense as they realize that the means for avoiding this undesirable outcome are decreasing in either quantity or quality. While uncertainty was key in the definition provided by Ortony, Collins, and Clore [59], they require that the uncertain event have consequences that are either highly desirable or undesirable to the audience.

Branigan ties desirability of outcome in with a *disparity in knowledge* between the audience and characters [12]. Authors create suspense by providing the audience with more knowledge than the characters possess, particularly about the possibility of undesirable outcomes (which the characters may or may not be aware of). As audiences learn more about an undesirable outcome, the more suspense they feel, so long as the characters are not also gaining the requisite knowledge. Like Abbott, Branigan notes the close relationship between suspense and surprise. While suspense is generated when the audience knows more than the characters, surprise is created when characters possess more knowledge than the audience.

A final component for suspense is *affinity* for the character in question, typically the protagonist. Zillmann requires that the uncertain and undesirable outcome affects a protagonist for whom the audience has positive feelings in order for the audience to feel suspense [80]. Further, Comisky and Bryant's study of film viewers demonstrated that not only was suspense affected by the protagonist's chance for survival, but also that the degree of reported suspense increased with the audience's affinity for that protagonist.

It appears to be the consensus of narratologists, psychologists, and entertainment theorists that uncertainty of outcome, desirability of outcome, a knowledge disparity,

and affinity for the protagonist are the necessary components for producing suspense in a narrative. Foremost, uncertainty is present as a factor in suspense in each of these definitions. For Gerrig and Bernardo, the uncertainty lies in the number or adequacy of the potential escapes for the protagonists. For many of the other definitions, uncertainty simply means a question about which of two potential outcomes will come to pass. Zillmann de-emphasizes uncertainty in favor of desirability, but still maintains it as a critical element of suspense. Zillmann's requirement that the outcome be undesirable overlaps with the definitions provided by Tan, Ortony et al., and Gerrig and Bernardo. The existence of a disparity in knowledge between the audience and characters creates suspense by informing the audience of circumstances unbeknownst to the characters. Reversing the knowledge disparity produces audience surprise. Finally, the requirement that the audience have positive feelings for the protagonist (or the actor facing the uncertain and undesirable outcome) is present in the study by Comisky and Bryant and echoed by Zillmann's definition.

2.3 Suspense in Dramatis

While I have introduced several definitions of suspense thus far, this thesis adopts the definition provided by Gerrig and Bernardo [30]:

> Readers feel suspense when led to believe that the quantity or quality of paths through the hero's problem space has become diminished.

While the other definitions I have discussed provide additional detail, this particular definition can easily be connected to human problem-solving processes. Problem-solving is the foundation of many approaches within artificial intelligence, including planning, which will be described later in this chapter. Gerrig and Bernardo describe a reader evaluating a search space to solve a problem on the behalf of the protagonist. This suggests that this definition, unlike the others presented in this chapter, have a potential computational approach. Additionally, Gerrig and Bernardo evaluated

their definition of suspense by testing different versions of the same story on human readers, providing some insight on how to evaluate a computational representation of their model.

To understand the approach to suspense taken in Dramatis, it is necessary to review some of the details of Gerrig and Bernardo's work. Gerrig and Bernardo use a scene from Ian Fleming's James Bond novel *Casino Royale* as an example. In a pivotal scene, James Bond, and readers thinking on his behalf, are attempting to traverse a search space from the current state (where Bond has a gun pointed at his back), through a series of intermediate states, to a goal state (any state where Bond's life is no longer in danger). However, the difficulty is that the set of solution paths through those intermediate states are limited. In the particular case of *Casino Royale*, Fleming explicitly suggests solutions (assistance from friends or casino staff) to the reader via Bond's internal monologue. Fleming removes these paths, thereby pruning the search space, and reducing the quantity of available solutions. Gerrig and Bernardo note that authors can also manipulate the readers' (or problem solvers') beliefs about the quality of solutions by making it appear as though actions—that is, transitions between intermediate states—are unlikely or too challenging to be viable. While such solution paths are not pruned from the search space, they are made to appear so costly as to not be worth investigating further. Thus, by actually reducing the search space, or by apparently making areas of the search space appear extremely costly, authors can manipulate readers into feeling suspense.

Artificial intelligence systems evaluate search systems in much the same way as Gerrig and Bernardo suggest happens while feeling suspense. Consider the James Bond example again. Readers are attempting to traverse the search space from the current state of the story to any state where Bond is safe. This describes a planning problem in artificial intelligence (For more detail on planning, see Section 2.6). An intelligent planner would be looking for a sequence of actions to get Bond from the

current situation to a better one, where each action that Bond takes comes with some cost. In order to help the reader/planner, Fleming suggests solutions, thereby reducing the complexity of the search space. However, once Fleming makes clear that these solutions will not actually work out, the reader/planner must start over again, with one less plan available. Further, Fleming can reduce the quality of a solution/plan by making them appear so costly that the reader/planner gives up and tries another strategy. Thus, Gerrig and Bernardo are describing a planning problem. The audience is evaluating a set of states, and trying to transform the current state of the story into a better one for the protagonist.

Conversely, other definitions are less indicative of a computational representation. Gerrig and Bernardo describe a set of actions on the part of audiences, where others, such as Abbott or Tan, merely describe suspense as a narrative procedure—a tool in the author's toolbox. Ortony et al. describe suspense as a mixture of uncertainty, hope, and fear. However, hope and fear are not easily measured in audiences, and therefore not easily formalized. By describing suspense in the context of the story world and how readers perceive it, Gerrig and Bernardo provide a definition for suspense that has a clear computational analogy.

While Gerrig and Bernardo's definition of suspense does include uncertainty and undesirability of outcomes, as well as the knowledge disparity between the characters and audience, it does not explicitly address character affinity. Like hope and fear, affinity for a character is not easily measured, nor is it a trait that can be expected to be constant from reader to reader. As a result, I will not take character affinity into account in my suspense model. While ignoring character affinity may limit the model, it is clear from their ubiquity in the definitions of suspense that uncertainty and outcome desirability were the most important factors in recognizing suspense.

2.4 Models of Narrative Inference

In this section, I describe the processes that readers use to comprehend stories, as well as computational approaches to modeling this comprehension process. These processes will inform the suspense model, particularly the memory model component, which tracks reader attention to the various items and characters in the story being read.

Throughout the reading process, readers make inferences about aspects of the story that have not been explicitly stated in order to make sense of the narrative [34]. Some inferences—*online inferences*—can be made with little effort while reading. Online inference tasks include recognizing characters' goals, identifying the reference of a pronoun, recognizing the causal antecedents of a new event, and identifying the theme of the story. Conversely, some inferences – *offline inferences* – can only be made when the audience is afforded time to reason. Offline inference tasks include recognizing the causal consequences of an action and inferring the plan of action used to achieve the current state of the world. Not all inferences can be easily classified as online or offline. Graesser et al. note that it is unclear whether divining the intent of the author in a passage should be considered an online or offline inference. Further, they point out that this is not necessarily a binary state. Rather, there is a spectrum, and references described as online might better be described as more likely to be generated online than offline.

The Event Indexing (EI) model, proposed by Zwaan et al., is a psychological model of a reader's conceptualization of a story as it is being read [81]. The story, and each event within the story, is tracked along five dimensions: the protagonist, temporality, spatiality, causality, and intentionality. Temporality refers to the time in which a story event occurs. Spatiality refers to the location of an event. Causality refers to the causal relationships between events, while intentionality refers to the event's relationships to the goals of characters, particularly the protagonist's goals. The

model maintains indices for each of these dimensions, and the values of these indices change when the corresponding information changes within the story. Further, Zwaan et al. propose that the reader has an activational model of story information. The activation of a particular story element is tied to how recently that element was seen, as well as the activation of highly related concepts. When an index value changes (e.g., a flashback changes the temporality of the scene, a change in location, or the protagonist accomplishing a goal), the previous value for that index is deactivated, and the new value activated in its place. When activation changes, a reader's ability to process information is slowed.

Niehaus' Inferences for Extending Recall (INFER) system sought to model narrative focus as a means of determining what inferences, whether online or offline, a reader would be able to make while reading a story [57]. INFER models causal inferences, where readers discern missing events of a narrative sequence, and intentional inferences, where readers discern a character's unstated goal from their actions. These inferences are made easier when relevant actions and objects are at the forefront of a reader's mind. INFER uses a modified EI model (MEI) to track the activation of various story elements in order to measure the salience of these elements in a reader's mind.

Cardona-Rivera et al. describe Indexter, a similar model of the reader based on the EI model [16]. Where MEI uses a spreading activation model to capture the salience of story events, Indexter's model is based on an expanded version on the IPOCL plan structure [66]. Unlike other planning domains, IPOCL plans include structures for character intent, which allows for mapping to the intentionality index of EI. Additionally, each plan operator in this expanded IPOCL representation specifies a time and location, while the plan's initial state indicates a story's protagonist. Finally, IPOCL provides a representation for the causal relationship between events, allowing Indexter to map the causality index from EI. Thus, Indexter is able to map

all five indices from EI into the IPOCL plan structure. Indexter adopts the binary representation of each index proposed by Zwaan for the EI model, while discounting the activation-based approach that Zwaan also suggested. Dramatis expands on MEI, rather than Indexter, because of its ability to track the salience of specific story elements using the activation model.

2.5 Spreading Activation Theory

As stated above, Niehaus' MEI memory model uses a *spreading activation* model in its representation of a reader's conceptualization of a story as it is being read [57]. Spreading activation is a common model for human memory. In spreading activation models, including MEI and Anderson's Adaptive Control of Thought (ACT) theory [2, 3], an element's activation determines how likely a person is to retrieve that concept from long-term memory, and how quickly that retrieval will occur. Elements are interrelated, such that the strength of association between concepts affects whether one concept is primed by another. In a spreading activation model, the activation of an element that was recently observed is passed on to other concepts. How much of the activation is spread depends on the strength of the association between the elements. We can therefore view the model as a weighted graph, where nodes represent the elements in memory and edges represented the associations between elements.

The time required to retrieve a concept from memory is inversely related to its activation level. Additionally, the probability of successful retrieval increases with activation. In ACT theory, an element's activation is a function of three components of the concept network [3]:

- The concept's *base-level activation*
- The weights given to the inputs that are triggering retrieval

- The strength of the associations between those inputs and the concept in question

Baseline activation increases when a concept is used repeatedly. Weights are given to inputs that prime retrieval from memory. As a simple example, we would give a weight of 1 to words that were recently seen, and weights of 0 to other words.

Different models use different functions for spreading activation between elements. In general, activation is spread according to the strength of the association between two elements. When an element is active in memory, such as when it is read, elements that are connected to the primed element gain activation in proportion to the strength of the association. This increase in activation can be further spread to elements that are further away from the original node, until the change in activation tapers out and the network stabilizes.

Unlike ACT, the spreading activation model used for the MEI memory model does not provide elements with a baseline activation. Instead, time is an implicit factor in determining the strength of association between two elements, and therefore the activation of those elements. Each node is added to the graph when it is first observed in the story, and given an initial activation of 1.0. Activation spreads to other nodes according to the weights on the associating edges. This process iterates until the activation of each element in the network has stabilized. While an ACT-like spreading activation model could be used for the memory model in Dramatis, I chose to use the MEI model because of its relationship to stories, and its focus on elements such as event causality and temporality. These features could be imposed on to an ACT-based memory model, but the MEI model provides rules for creating the network in a framework specifically intended for story understanding.

2.6 Planning

Planning is a technique commonly used in both story generation and story understanding. Stories are sequences of actions, and some sequences of actions may be interpreted as plans. Thus, in certain domains, by generating a plan, one has also created a story. Additionally, one can use planning as a means of filling in gaps in stories. Recall from Section 2.3 that Gerrig and Bernardo's definition of suspense can be described as a search space. Readers plan on behalf of the protagonist in order to help him avoid some negative consequence. The success of this attempt at planning affects the amount of suspense a reader feels. Thus, Dramatis uses planning in its approach to calculating the quality of plans available to the protagonist.

Planning is the process of searching for a sequence of operators that transform some initial state to a state in which some goal has been achieved. (Note: All planning definitions in this section are influenced heavily by Ghallab et al. [33] and Hogg [38]).

Definition 2 (Plan): A *plan* is any sequence of actions $\pi = \langle a_1, \ldots, a_k \rangle$, where $k \geq 0$ and actions are instantiations of plan operators [33].

The sequence of actions in a plan is only valid for the specific initial state and goal predicate specified prior to beginning the search for a plan. Thus, a planning problem may be defined as the initial state, the set of propositions that must be true in the goal situation, and the planning domain. A planning domain is made up of a set of states, a set of operators, and a function for transitioning between states. The set of states in the planning domain is the set of all possible states that could exist in the world given the propositions used within the planning problem. Operators are functions that transform the world from one state to another and represent actions in the real-world. The state-transition function defines what the next state will be given the application of an operator to a particular state.

Definition 3 (Planning Problem): A *planning problem* is a tuple $P = \langle \Sigma, s_0,$

$g\rangle$, where Σ is a planning domain, $s_0 \in S$ is the initial state, and g is a set of propositions (referred to as *goal propositions* that must be true in a state for a state to represent a goal situation. For any state $s \in S$, if $g \subseteq s$, then s is a goal situation.

Definition 4 (Planning Domain): A *planning domain* is a tuple $\Sigma = \langle S, O, \gamma \rangle$, where S is a set of states, O is a set of operators, and γ is a map $(S \times O) \rightarrow S$ is a state-transition function.

An operator contains a name, a set of parameters, and three sets of propositions. The name is unique to all of the operators in the planning problem. The first set of propositions represents the preconditions—the propositions that must be true in the world before an operator can be applied. The second set is the added effects—the set of propositions that will be added to the world state after the operator is performed. The third set of propositions is the deleted effects—the set of propositions that will be removed from the world state after the operator is performed. The parameters are symbols that describe elements of the world that will be referenced by the proposition sets. Uninstantiated operators use variables to represent the characters and objects that are parameters to the operator. The definition of an operator provided below is based on the STRIPS [25] model of planning.

Definition 5 (Operator): An *operator* is a tuple $o = \langle$ n, PARAMS, PREC, ADD, DEL $\rangle$, such that n is the unique name of the operator, PARAMS are the variable symbols that appear within o, PREC is a set of propositions representing the preconditions of o, ADD is the set of propositions being added to the world state as effects, and DEL is the set of propositions being removed from the world state as effects.

We can describe a pair of operators as being causally linked if one or more effects

of the preceding operator in the plan is also a precondition of the other operator. Let us define a causal link as follows:

Definition 6 (Causal Link): A *causal link* is a tuple $\langle a_i, a_j, p \rangle$, where a_i and a_j are actions in the same plan, a_i precedes a_j in that plan, and p is a proposition, such that p is an effect of a_i and a precondition of a_j.

2.7 Aesthetics in Computational Creativity

Computational creativity refers to the area of research into developing intelligent systems that are capable of generating creative artifacts or performing creative acts. Such systems have been developed in a variety of domains, including stories, poetry, art, music, and design, among others. Several computationally creative systems have sought to include a model of aesthetics as part of the creative or evaluative processes performed by the systems. In this section, I briefly describe the role of aesthetics in computationally creative systems in non-narrative domains. In the subsequent section, I will go into greater detail about story generation and describe a number of systems in that domain that use aesthetic models.

Colton et al. developed the FACE model, intended to describe the set of generative acts that a computationally creative system could perform as part of a creative act [21]. Two of the possible generative acts include aesthetics, where a system may contain an *aesthetic measure*, or may contain a process for generating aesthetic measures. Colton et al. define an aesthetic measure as a function that maps from a procedure for generating a creative artifact and its output to a non-negative real number. Other components of the FACE model include the procedure for generating a creative artifact and language that describes that artifact or process. Colton proposes that computational creativity researchers begin evaluating creative systems according to the impact of the systems, including the development of aesthetic measures, rather than evaluating the artifacts of such systems according to pre-defined

aesthetic measures.

Datta et al. [24] developed a classifier and regression model for rating the aesthetics of photographs from a corpus of images on a photography website. Each photograph in their corpus had been rated by website users in the categories of aesthetics and originality. The researchers found a strong correlation between the scores for originality and aesthetics, and chose to narrow their focus to aesthetics only. The classifier distinguishes between photographs that had been rated highly for aesthetics and those that had been rated poorly. The classifier was developed from a set of 56 features of the photographs. These features include the use of light and color, the saturation of the image, similarity to other images in the corpus, and use of canonical photography rules such as the "rule of thirds." These features were reduced to a set of 15 features that achieved good accuracy and precision at classifying the photos into "low" or "high" ratings. The classifier improves as the numerical distance between "low" and "high" ratings increases, as well as when the number of independent raters for the photographs increases. While Datta et al. do not discuss the implications of this classifier from a creative perspective, it is not difficult to imagine using this classifier as a heuristic for computer-generated images meant to simulate photography. Additionally, the classifier is a model of human aesthetics. It can function as a critic of human-generated photography, as its ability to classify photographs is derived from known opinions of aesthetics from human raters.

Reich implements an aesthetic model in BRIDGER, a system that assists in the preliminary phases of bridge design [64]. Reich notes that most design literature emphasizes the creation of artifacts that meet some set of functional requirements and ignores those elements which influence the aesthetic quality of the artifact. The aesthetic model in BRIDGER recognizes that aesthetics are dependent on the culture and the observer. Therefore, the model does not make use of fixed rules of aesthetics, treating aesthetic knowledge as guidelines that can be ignored or adapted. While the

BRIDGER system is limited to a specific type of bridge, the general model of design aesthetics can be transferred to other areas of design.

2.8 Aesthetic Models in Story Generation and Interactive Narrative

Whether it is possible for an intelligent computational system to generate an aesthetically pleasing narrative has remained an open question for more than thirty years. Computational methods for story generation can generally be broken down into search-based approaches (e.g. [52, 45, 61, 66]) and adaptive approaches (e.g. [73, 60, 32]). The history of story generation is outside the scope of this thesis, although Gervás [31] provides a detailed history of the research area. This section will briefly cover story generation, and then focus on story generation systems that implement some sort of model of aesthetics.

Search-based story generation systems create stories by exploring the set of possible sequences of actions, typically comparing the generated story against some heuristic of quality. Planners are used in some search-based story generation systems to generate stories, where a completed plan represents a full story. Search-based story generation is exemplified by early story-generation systems, such as TALE-SPIN [52] and Universe [45], as well as more recent approaches such as IPOCL [66] and the generator for interactive storytelling developed by Porteous and Cavazza [61].

In contrast to search-based approaches to story generation, adaptive story generation systems (e.g. Minstrel [73], MEXICA [60]) use their knowledge of other stories to recombine and modify these stories into new ones. In some cases, these systems use analogical reasoning to import new concepts into stories.

Interactive narrative is a form of story generation wherein the audience/user influences the narrative by taking actions in the story world [65]. Typically, the user takes on the role of the protagonist. However, other approaches to interactive narrative include commanding computer-controlled characters, or manipulating the story

world as an observer rather than a character. The user's actions have the power to affect the direction or outcome of the story as it unfolds.

Recall that narratologists distinguish between the events of the story world (the *fabula*) and the order of their presentation (the *sjuzhet*). While many story generation systems work to generate the entire fabula, it is also possible to generate a story by taking a fabula as input and generating a particular sjuzhet. In the remainder of this section, I describe two systems that generate sjuzhets according to a model of aesthetics, followed by several systems that generate fabulas according to aesthetic models.

2.8.1 Story Sjuzhet Generation with Aesthetics

Suspenser is one of several story generation systems that includes some model of aesthetics. Suspenser [19] computationally attempts to find a suspenseful telling of an existing story. Suspenser takes a *fabula* as input with the goal of identifying the most suspenseful *sjuzhet*. The system first identifies the skeleton story, based on what it judges to be the most important events of the story, where importance is tied to the causal connections between a given event and the other events of the story. Once the skeleton has been established, Suspenser identifies candidate events to be added to the story in order to produce the greatest suspense in a reader. Suspenser works from Gerrig and Bernardo's definition of suspense – audiences feel more suspense when the quantity or quality of escapes for a protagonist are reduced. Given this approach, Suspenser generates all possible plans a protagonist might have and takes the ratio of failed plans to successful plans. As the ratio increases, the suspense level increases as well. While the definition of suspense is supported by psychology literature, this implementation of the definition is problematic. All failed plans count towards the ratio, but failed plans are not produced only by a world state that has limited options for success. Failed plans could be produced because of poorly defined

planning operators, problems binding operators to literals, or other issues inherent to the nature of planning rather than the story scenario. This assumption that failed plans are a result of the world state rather than the planning problem definition makes this approach a poor computational implementation of Gerrig and Bernardo's definition of suspense.

Just as Suspenser finds the most suspenseful *sjuzhet*, Prevoyant [5] uses a computational model of flashback and foreshadowing to produce a *sjuzhet* intended to elicit feelings of surprise in human readers. Flashback and foreshadowing are narrative techniques that reveal events of the *fabula* out of temporal order with nearby events. Prevoyant uses these techniques to foreshadow some outcome by revealing an outcome without making entirely clear how it could occur and then using flashback to reveal an event that initiated the already-seen outcome. This generates surprise in human readers because an outcome is provided without initiating events. The flashback then resolves the questions that arise from seeing the surprising events out of order. Prevoyant begins its process by selecting an event for flashback. It then identifies an important outcome and selects an event that has a direct causal link to that outcome. If this event can be removed from the story such that it only affects the causal link to the outcome, it is used for flashback. Prevoyant then selects some aspect of that event – a character or an object – to be introduced earlier in the story, as a means of foreshadowing. There are two important things to note about Prevoyant's process. First, surprise is a binary state within Prevoyant. There are no degrees of surprise or unexpectedness; rather, unexpectedness is defined as the reader's inability to find a plan explaining the outcome without the events of the flashback. Second, like Suspenser, Prevoyant receives the *fabula* as input, finding the most surprising *sjuzhet* without generating any new content.

While Suspenser and Prevoyant consider emotional aesthetics, other systems have

considered non-emotional aesthetic qualities, such as novelty [60] and character believability [66]. While these approaches are inherently aesthetic considerations, they do not address the question of how to elicit emotional responses from human readers. Suspense, curiosity, and surprise produce satisfaction and enjoyment from readers [13]. Attributes such as novelty and character believability, while certainly improving the overall quality of the generated stories, are not associated with eliciting emotional responses.

2.8.2 Story Fabula Generation with Aesthetics

While Suspenser and Prevoyant model suspense and surprise as part of the creation process, other story generation and interactive narrative systems have modeled the tension level within the story. While tension is frequently tied to suspense, none of these systems are trying to explicitly make the story more suspenseful. In general, each system attempts to incorporate dramatic arc in the interaction or the story generation process.

The MEXICA story generation system [60] models reader tension as a measure of how satisfactory the state of the story is. Certain events in MEXICA define an increase or decrease in tension as an effect of the event. For example, if a character is attacked, one postcondition of the event is an increase in tension. Similar effects exist when characters die or are taken prisoner. In these cases, the existence of tension is maintained until the characters are no longer in the same location or some other event postcondition deactivates the source of tension (such as a prisoner being freed). MEXICA also infers certain postconditions based on the world state. After each action, MEXICA checks for emotional conflicts and love triangles. The existence of either of these increases the tension level. No matter the source of the tension, the amount of the change in tension for a given event is defined in advance by the user of the system. MEXICA tracks the tension over the course of the story and compares

the changes in tension to other stories that it knows. MEXICA considers new stories more interesting when the Tensional Representation of the new story is similar to its knowledge of tension in previously generated stories. This process is intended to allow MEXICA to produce novel stories that remain well structured.

Porteous et al. [62] describe an interactive storytelling (IS) system inspired by Shakespeare's *Merchant of Venice*. While using planning for the narrative generation aspects of the IS task, one of the heuristics used to identify good plans is plan dynamics. As part of their consideration of plan dynamics, their system considers tension, in order to make stories conform to a typical Aristotelian dramatic arc. Planning operators are tagged with information regarding their contribution to the overall tension level of the story. Porteous et al. also allow authors to establish *constraints* to control for event-ordering and pace. These constraints are also tagged with tension levels that can be set by the user. An additional component to their IS system is the Narrative Arc Window, which gives users a visual representation of the tension levels attached to the authored constraints in the form of an Aristotelian arc. Users can draw or manipulate the dramatic arc and reorder the authored constraints. The IS system applies the user-provided information in order to find a best-fit tension arc in terms of the ordering and pace of the authored constraints.

The interactive drama, Façade [50], tracks tension as a factor in managing the interactive narrative. The Façade drama manager has an ideal tension curve and probabilistically changes the tension in the narrative to try to match the ideal curve. Tension in Façade is measured on a scale from 1 to 4. The tension level is a guide for the drama manager, while also affecting whether the player can have affinity for both of the protagonists in the drama. Each event in the story has a number of possible presentations based on character affinity and the tension level. The drama manager selects a presentation based on how well it fits the scene so far and the ideal tension level at that point of the story. Regardless of the tension level, each possible

presentation conveys roughly the same information, with the dialog and some other details varying depending on the tension.

Search-based drama management (SBDM) and declarative optimization-based drama management (DODM) are techniques used to make Tea for Three [74] and drama-managed versions of the Anchorhead interactive fiction [55, 56]. In these general approaches to drama management, authors specify an evaluation function in order to rate the quality of sequence of plot events. One of the main assumptions of SBDM and DODM is that this evaluation function can represent an author's aesthetic. Thus, the drama manager in an SBDM/DODM system is attempting to guide the user toward a story that meets the aesthetic goals of the author. The evaluation function in the drama-managed version of Anchorhead, regardless of technique, values several factors including consistency of location, actions fitting with character intentions, and how much freedom the user has in making choices. The role of the drama manager in Anchorhead in meeting the aesthetic goals of the author is comparable to two other systems described here. The drama managers in Façade and *Merchant of Venice* try to match ideal tension curves, which were similarly specified by authors, representing their aesthetic goals.

Szilas' interactive narrative architecture, IDtension, includes a model of the user [70]. This model is used to assess the set of actions available to the narrative sequencer at any given time, according to how the user of the interactive narrative would perceive the possible action. This perception is based on a number of narrative effects, which include but are not limited to character motivation, character ethical consistency, relevance to previous actions, and, most importantly, conflict. Szilas particularly targets ethical conflict, wherein characters must make decisions with the potential to violate their own values. Despite the inclusion of this model, Szilas states that the narrative generated during architecture evaluation lacked "dramatic intensity" [70] and that the attempts to increase the audience (or user's) emotional involvement was

insufficient for good narrative [71].

GADIN, another interactive narrative system, introduces dramatic tension by incorporating dilemmas which users must overcome, according to the belief that "drama is conflict" [7]. The system maintains a set of clichèd dilemmas, where making a decision has instantaneous effects on either the user's own utility, or that of another character. These prototypical dilemmas include betrayal and sacrifice, among others. When the plot is created, dilemmas are inserted to increase dramatic tension for the user. GADIN has no explicit model of audience or user interest. Instead, it operates from a designer-defined set of interactions, based in the idea that conflict builds tension and yields interesting drama.

Bailey proposes a story generation system based on a model of the responses of a typical reader [6]. Bailey models response in terms of the expectations and questions generated by the reader. The story generator is guided by a heuristic that seeks to achieve optimal "storiness," a trait defined by the patterns of expectations and questions according to the reader model. Questions are considered more important to storiness, as sequences of expectations without obstacles would be uninteresting. Additionally, Bailey uses Freytag's triangle [27] as a basis in his heuristic. Like Bailey, Mawhorter and Mateas have introduced preliminary work toward a story generation system that uses a reader-model in the generation process [51].

Yu and Riedl [77] present a drama manager that takes player preference into account when guiding the story. The drama manager uses a model of the player, based on the ratings of past players for similar plot points. The drama manager is thus working with two models of aesthetic: the players' own preferences for stories, along with the authorial model of aesthetic that went into the design of the story and the choices available to players.

Where many of the other systems modeled the aesthetic goals of the author or designer (e.g. Façade, Anchorhead, *Merchant of Venice*), Bailey, Mawhorter and

Mateas, and Yu and Riedl all focus on aesthetics from an audience perspective. Both Bailey and Yu & Riedl guide the generation and management task according to a model of the reader. Bailey uses a model of reader expectations and questions, while Yu and Riedl attempt to create a model of player preference to guide the narrative. By contrast, the other systems in this section are trying to match an authorial model of aesthetic, such as an evaluation function or an ideal tension curve.

Additionally, several of these authorial models of tension are hand-authored by the developer in ways that are inherently context-specific. In MEXICA, certain events are defined as altering the tension level as a postcondition. While this is acceptable within a localized domain (such as MEXICA's domain of traditional Mexican folktales), this approach cannot reasonably carry over into even moderately larger domains where tension may be context-dependent. One can imagine a scenario where the death of a character may not increase the tension depending on the other events of the story, while in MEXICA, such an event is guaranteed to increase the tension. In the case of Porteous et al.'s *Merchant of Venice* IS system, the authored constraints are tied to the specific story, rather than a domain of stories, making the domain dependence even greater. Tension ratings would have to be authored manually for each constraint in each interactive story. Façade suffers from the same problem. Façade contains multiple versions of the same sequence of events depending on the level of tension at the given moment. In each of these cases, the domain-dependency arises from tension levels that are authored in advance by users or developers. A better model of aesthetic would be domain-independent, such that specific events do not inherently affect the tension level in a pre-defined way.

I describe these story generation systems in order to reinforce the idea that determining whether a story is suspenseful, let alone creating a suspenseful story, is a difficult task. My system only provides a means of determining how suspenseful a story is. I do not attempt to generate suspenseful stories, although this model could

conceivably serve as an evaluation metric for a story generation system. I discuss the applications of Dramatis in story generation in Section 6.3.

2.9 Affective Models

Developing models of audience emotional responses requires modeling the affective processes that produce these emotions. Ortony et al. [59] describe a general theory of emotion based on the concept of *appraisal*. Appraisal describes the processes used by emotional agents, including humans, to determine the emotional significance of events. Under Ortony et al.'s (hereafter "OCC") theory of appraisal, the emotions felt by agent are influenced by that agent's plans, goals, and expectations. With this approach, two agents can observe the same event but feel different emotions depending on their other desires. OCC's appraisal theory has been a guide for a number of computational models of emotions, including Neal Reilly's Em architecture [54] and Gratch's Émile [35]. These examples focus on how emotional agents should interact with humans within a given educational or entertainment context. In contrast, my work focuses on the emotions of an audience that is not directly interacting with the narrative context.

OCC defines suspense as a hope emotion, a fear emotion, and a cognitive state of uncertainty. Hope and fear play into the character affinity aspect of suspense. We hope for some positive outcome for a character we like, but fear that the negative outcome will come to pass. While my work does not directly apply appraisal theory, it should be noted that the OCC definition of suspense can fit with the definition of suspense applied in this thesis. The requisite state of uncertainty is represented in my model of suspense as the reduction in possible escapes for the protagonist from the negative outcome.

Zillmann provides an alternate account of emotion, particularly with respect to drama. He describes affective disposition theory as part of an explanation of the

psychology of suspense in drama [80]. He states that audiences feel empathic responses towards agents for whom they have a positive affinity. When they have a negative affinity towards an agent, that empathic response is reversed. In the context of drama, Zillmann argues that audiences desire positive outcomes for liked characters and negative outcomes for those characters that are disliked [79]. Additionally, audiences fear the reverse – that is, negative outcomes for liked characters and vice-versa. Zillmann argues that suspense comes from the mere suggestion of negative outcomes. This parallels the OCC definition of suspense. Suspense comes from fearing negative outcomes while hoping for positive results for characters we like. While I am not aware of any computational models of emotion based on affective disposition theory, there is a strong link to the OCC model, which has been shown to be applicable in computational systems.

2.10 Categorizing Suspense

It is my opinion that, despite the common factors identified among the definitions of suspense in Section 2.2, suspense can be created in different ways. In this section, I will present several stories that I believe generate suspense in different ways. I will then categorize these types of suspense, using these stories as representative examples. The purpose of this task is to specify precisely what forms of suspense my model will be able to detect and measure. Figure 2 shows the categories I describe and the stories I use as examples.

2.10.1 Example Stories

Consider the following example stories:

- ***Casino Royale*:** A waitress delivers a drink to James Bond which has been poisoned. He realizes this fact, and attempts to cure himself of the poison. He first attempts to vomit the poison out, but this fails. With the help of MI6,

he locates an antidote and is forced to restart his heart using a defibrillator. In this example, suspense is generated from Bond's various attempts at curing himself which fail, forcing him to come up with new strategies, some of which viewers are familiar with because of their familiarity with spy movies.

- ***Rear Window*:** Jeff and Lisa suspect their neighbor, Thorwald, of murdering his wife. Jeff is in a wheelchair, unable to leave his apartment, so Lisa breaks into Thorwald's apartment alone to find evidence. Thorwald catches her before she can get out, while Jeff watches from across the street. The suspense in this story comes from the audience's affinity for Lisa, and the danger that the audience believes that Lisa faces at Thorwald's hands.

- ***Harry Potter and the Half-Blood Prince*:** Professor Dumbledore and Harry return to Hogwarts. Dumbledore initially sends Harry to get help from Snape, but tells him to hide when he hears footsteps. Before Harry can get help, Draco appears to attempt to assassinate Dumbledore. Readers feel suspense while reading this because they are aware of Draco's plan to kill Dumbledore, and because readers doubt that Snape will actually help Dumbledore, if he arrives.

- ***Toy Story 3*:** Woody, Buzz, and several other toys find themselves on a conveyor belt that is leading to a set of blades, and beyond that, an incinerator. The toys attempt multiple methods of escape, but the conveyor belt continues to move them closer and closer to the incinerator. Again, readers see the toys attempt solutions, fail, and come up with new ones, all while the conveyor belt moves them closer to imminent death.

- ***Friends*:** Two characters, Ross and Rachel, are clearly attracted to each other but are not in a relationship. Over the course of the series, various obstacles prevent the couple from coming together. Viewers feel suspense over the

uncertainty of the question of whether Ross and Rachel will get together.

- ***Final Destination*:** One character is given a premonition to avoid his death, which leads to him saving a group of people. Death then attempts to correct this by killing the other survivors. The audience is aware that Death is trying to kill them, but the means that Death will use are not made clear until the character's actual demise. The audience feels suspense as they wait to see if any of the characters will survive, or how Death will kill each character.

- ***The Bourne Identity*:** Jason Bourne is in a bank, and in danger of being captured by the police. As the police close in, he realizes he has no weapons and there are too many people to fight at once. As one of the officers grabs him, Bourne grabs a pen from a table and uses it to stab one of the officers. Viewers feel suspense because there is no obvious solution for Bourne. He overcomes the situation by using an object (the pen) in an unexpected way.

- ***MacGyver*:** In prototypical episodes of MacGyver, the titular character must improvise some means of escape or overcome some obstacle using only the objects that he has on him or are lying nearby. For example, MacGyver creates a bomb using chewing gum, a paperclip, and a pair of glasses. Like Bourne, viewers feel suspense because there is no obvious solution to the characters' predicament. Rather, he must use the objects around him in ways that viewers might not be able to predict.

2.10.2 Suspense Types

Consider the suspense that audiences feel while watching *Casino Royale*. Audiences are familiar with the typical events of spy movies, and as a result, they are able to predict upcoming events or outcomes on the basis of prior experience with similar narratives. Put simply, as audiences become familiar with a genre of stories, they

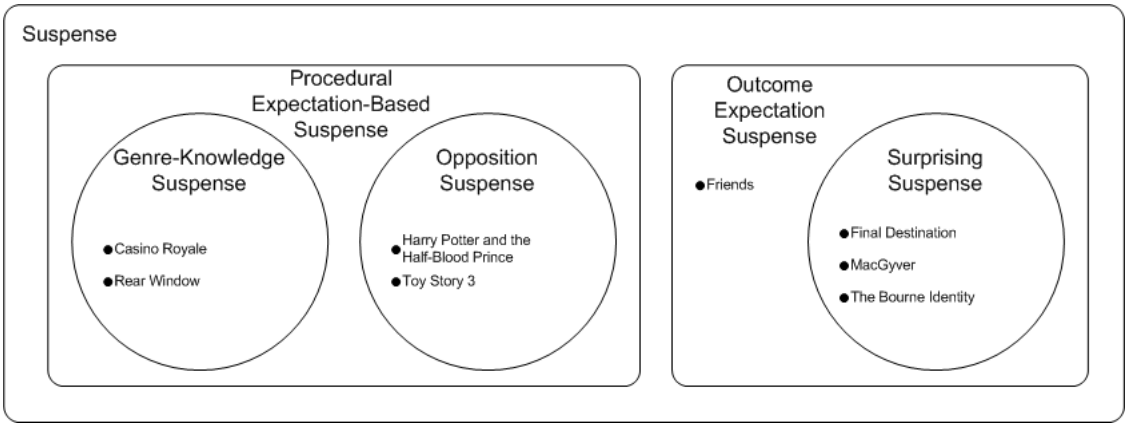

Figure 2: Categories of suspense and their membership

gain knowledge that explain the typical events and interactions of that genre. This knowledge allow audiences to infer what obstacles characters may face, as well as typical means of overcoming those obstacles. I refer to such suspense as *genre-knowledge suspense*. The suspense in *Rear Window* could be considered similar. Audiences are familiar with what happens when people break into others' homes, as well as what may happen if the homeowner returns. This allows them to predict upcoming events or obstacles.

The suspense generated in *Harry Potter and the Half-Blood Prince* is similar to genre-knowledge suspense, in that is related to the audience's knowledge. Rather than being aware of typical interactions of a genre, the audience is instead knowledgeable about the plans of the antagonists (in this case, Draco), because they have explicitly been given knowledge about these plans. I refer to this type of suspense as *opposition suspense*. As with genre-knowledge suspense, audiences in this instance have knowledge about what obstacles the protagonist will face. What has changed is the source of the knowledge. Rather than gaining this knowledge through experience with narratives or a particular genre, the author has provided the audience with the opposing plan. One could also consider the conveyor belt, and its movement, in *Toy Story 3* as an opposing plan.

In each of the above conditions, audiences are cued to expect some sequence of

events, either because of their own knowledge about a genre of stories, or because knowledge of this sequence has been provided by the author. As a result, I refer to the combined space of genre-knowledge suspense and opposition suspense as *procedural expectation-based suspense*.

Unlike the above examples, there are cases where an audience is inclined to expect, and possibly desire, a particular outcome, but the knowledge of how to achieve that outcome is not provided. Let us refer to this type of suspense as *outcome expectation suspense*. Consider the *Friends* example above. In these stories, suspense is generated from a "Will They/Won't They" question on the part of the audience – that is, will the protagonists come together as a couple, or will the conditions of the story prevent this outcome? Romantic comedy audiences expect and desire the *They Will* outcome. However, audiences are not provided with the path to this outcome in advance within the story, nor do they possess the necessary genre knowledge. Where in the previous category, this outside knowledge provides information about what negative states are upcoming, characters in romantic comedies are in the negative *They Won't* state by default. What limited knowledge the audience does have about the genre may be more abstract. Some obstacle will keep the couple apart, and attempts to overcome these obstacles will fail at reducing the distance between the characters. By contrast, if characters were initially in a *They Will* state, then it might be possible to forecast a series of events that lead to a break-up (a *They Won't* state), and generate plans to prevent that break-up from coming to be. The lack of specific knowledge about the genre or the characters' plans when starting in the *They Won't* state, as romantic comedies typically do, separates outcome expectation suspense from procedural expectation-based suspense.

Similarly, consider the possibility where the audience is entirely aware of the outcome, and the suspense is generated solely from their lack of knowledge about the series of events that will lead to that outcome. For example, in the *Final Destination*

series of films, the audience is aware of the characters' impending death, but there is nothing to indicate exactly how the characters will die, nor how the characters can prevent it. In most cases, guessing would be impossible, as the means of death tend to be unexpected. In some cases (e.g. *MacGyver* and *The Bourne Identity*), the creativity involved on the part of the character in generating the outcome may be so surprising that the audience had no means of generating the same solution as the character. In each of these examples, audiences are induced to feel suspense about an outcome, or perhaps even a path to an outcome, but without any knowledge as to what that path may entail. I describe this type of suspense as *surprising suspense*, as the total lack of knowledge as to the specifics of the problem or the resulting solution leads to a surprise for the audience, regardless of the outcome. One could consider this set to be a subset of outcome expectation suspense, as both rely on some knowledge of the outcome with limited knowledge as to the solution path.

Gerrig and Bernardo's definition of suspense requires a space to search. Stories that make use of procedure expectation-based suspense provide that search space, either by informing the audience of opposing characters' plans, or by being part of a genre that audiences are familiar with. By contrast, stories using outcome expectation suspense do not provide audiences with this search space. Because the selected definition requires this search space, Dramatis specifically models procedural expectation-based suspense, and is not capable of modeling outcome expectation suspense. When genre knowledge is too abstract (e.g. *Friends*) or there is no realistic way of determining a solution (e.g. *MacGyver*), it is not possible to evaluate the search space on behalf of the protagonist, as Gerrig and Bernardo suggest. Thus, by selecting Gerrig and Bernardo's definition of suspense, I have limited Dramatis to stories that use procedural expectation-based suspense.

CHAPTER III

DRAMATIS

A constant problem in the development of story generation systems has been the difficulty in giving such systems a sense of aesthetics. These systems are unable to view artifacts from a human perspective, preventing accurate assessments of aesthetic and quality. In order to improve our attempts at story generation, we need to develop models of human aesthetic. Initial steps toward these models should focus on single attributes (e.g. suspense, surprise, curiosity) rather than a broader, all-encompassing model of human aesthetics. Suspense is a prime candidate for modeling, as it has been studied by scholars in an array of fields, including narratology, psychology, and entertainment theory. This work seeks to develop a computational model of suspense, based on understandings of suspense from each of these fields, focusing on the domain of stories. Successful work in this vein could lead to further progress in human aesthetics in computational systems in a multitude of domains.

3.1 Gerrig and Bernardo's Suspense Model

Developing a computational model of suspense requires a clear definition of the suspense phenomenon, which provides guidelines for the architecture and functionality of the model. In Chapter 2, I described several definitions of suspense from a variety of fields before ultimately settling on Gerrig and Bernardo's definition [30]. This definition has the benefit of being specific enough so as to imply a computational definition, while also covering many of the concepts proposed in other definitions. For ease of reference, I reprint that definition here:

> Readers feel suspense when led to believe that the quantity or quality of

paths through the hero's problem space has become diminished.

As described in Chapter 2, this definition of suspense describes the evaluation of a search space. Gerrig and Bernardo describe how readers, on behalf of a protagonist, attempt to traverse a search space from the current state of the story world to some goal state, where an imminent undesirable outcome has been avoided. Authors induce suspense by manipulating the audience's beliefs about the quantity or quality of solution paths in this search space. They can reduce quantity of paths simply by closing doors that had previously appeared open to the protagonist. Reducing the quality of solutions is possible by making actions in the search space—that is, paths between the initial state and the goal state—become too costly, or by making the actions, or the whole path, less likely to succeed. It is also possible to reduce the quality of a path by requiring more actions in that path, thereby increasing the cost. Thus, audiences can be manipulated into feeling suspense by changing the nature of the search space.

3.2 Reformulation of Gerrig and Bernardo's Definition

Unfortunately, Gerrig and Bernardo's model is computationally intractable for three reasons. First, a simple count of the paths that end in failures makes a wrongful assumption that all paths that are not part of a possible solution are the result of the story state, and not the result of how the planning problem was defined. Suspenser [19] takes this literal view of Gerrig and Bernardo's definition, measuring the rate of planning failures, but not distinguishing between all of the ways that a node in the search space can be terminal, but not a goal situation. Planning problems can reach terminal nodes in the search space by expanding nodes that have already been visited, by failing to bind operators to symbols, or by reaching states where no other operators in the planning domain are applicable. However, none of these reasons for failing to find a particular solution are necessarily indicative of high

suspense levels. Second, Gerrig and Bernardo's model seems to suggest that humans regenerate the search space repeatedly, considering suspense perpetually. However, the definitions of suspense that I outlined in Chapter 2 showed that human audiences do not feel suspense unless they are somehow induced to expect some undesirable outcome or *failure*—that is, something that goes wrong for the protagonist, like a gun being placed against James Bond's back. Additionally, regenerating the search space requires considering the causal consequences of story events, which is typically an inference that can only be made during offline processing [34]. Because human readers are not capable of frequent offline processing, a computational model of suspense should not regenerate the search space of the story repeatedly. Rather, the model should only do so when there is reason to perform the computation (i.e., when the audience has been cued to expect a failure). Third, I assert that there is no evidence that humans generate the entire search space, and thus all possible paths to failure or escape, in part because humans are resource-bounded.

Given these insights into suspense and computation, I introduce the following reinterpretation of Gerrig and Bernardo's description of suspense as search space, as guidelines for producing a computational model of suspense: Given the belief that a character can face a negative outcome, one can assume that this outcome will occur and search for the single most likely plan in which the protagonist avoids this outcome. I refer to such a plan as an *escape plan*. Furthermore, Gerrig and Bernardo refer to the "quality" of paths without being precise as to how quality is measured. I define the quality of an escape plan as its perceived likelihood of success. The use of *perceived* likelihood, rather than *actual* likelihood, allows us to consider the ways that authors can make solution paths appear more costly than one might originally anticipate. Further, by using perceived likelihood, I can account for some of the disparity in knowledge between the characters in the story and the audience planning on their behalf. Gerrig and Bernardo suggest that authors manipulate

reader perceptions of the story state in order to manipulate suspense. There is a known relationship between reader memory and the perceived likelihood of potential events. Specifically, the perceived likelihood of an event is correlated to how easily that event can be retrieved from memory. Humans consider the first thought they retrieve from memory to be the most likely thing to happen [48]. This would suggest that the first escape plan generated would be the one that a human reader would consider most likely to succeed, regardless of any objective measure of likelihood or quality. A model of reader memory that takes salience and connectivity into account would allow us to ascertain what plans a human reader would consider to be high-quality plans, primarily because of the relative ease of conceiving those particular escape plans as opposed to the alternatives. One significant benefit of this approach is that it requires only one plan to be found, without necessarily generating the entire search space. Alternately, one could assume that human readers would consider some number of plans that can be retrieved in some limited amount of time. Thus any plan that meets some pre-determined threshold of salience in the reader's memory would be an acceptable escape plan. Additionally, the model does not require repeated regeneration of the search space, because the search occurs only when the audience expects failure. Finally, by searching for a single escape plan (or even a limited set of salient plans) rather than the total space of plans, we avoid the problematic assumptions of counting planning failures as a means of calculating suspense, as applied by Suspenser [19].

As mentioned above, there is an established link between reader memory and the perceived likelihood of events. Given the importance of perceived likelihood to my reformulation of Gerrig and Bernardo's description of suspense, my model of suspense must take reader memory into account. Specifically, I note two concepts related to reader memory that are necessary to my model of suspense. First, readers will have stronger memory for story elements that they have seen recently, and to a

lesser degree, story elements that are strongly connected to those recently observed elements. Secondly, because the quality of an escape path is more about the perceived quality on the part of the reader rather than an objective measure of its quality, reader memory must be able to indicate the perceived quality of a character's plan, or at minimum, the steps or story elements of a character's plan.

This reinterpretation of Gerrig and Bernardo's definition of suspense helps reduce the complexity and intractability of the original definition. While planning, a process that occurs frequently, is a **PSPACE** problem, there is no longer a need to generate the entire search space, which would be intractable in any sufficiently complex domain. Additionally, the complexity is reduced by searching for a single plan, rather than the space of all possible escape plans and all planning problems that fail to find solutions. In the following sections, I introduce strategies that I use to further restrict the search space by using domain knowledge, collected from human readers, to identify the goal situations that should be targeted in the escape planning process. Without access to this type of knowledge, Dramatis would be blindly planning for a situation in which the protagonist escapes. Thus, while the escape planning problem remains a computationally complex task, I have reduced the overall complexity by limiting the number of necessary solutions and limiting the breadth of planning required by increasing the information available in the search process.

The suspense model that I describe in this chapter is not able to handle all types of suspenseful stories. In Section 2.10, I proposed a taxonomy of suspense. Dramatis is capable of handling a type of suspense that I describe as *procedural expectation-based suspense*. This form of suspense is supported by detailed knowledge on the part of the reader about either the genre of the story, or the plans in the characters in the story. By comparison, Dramatis cannot measure the suspense in stories where the suspense is based on an expected outcome where the expected events on the way to that outcome cannot be anticipated.

3.3 Computational Representation of Gerrig and Bernardo

My proposed model, Dramatis, calculates suspense for a given story according to the reinterpretation of Gerrig and Bernardo's definition of suspense presented above. Dramatis reads in a story in a symbolic-logic format, tracking the story state as it reads. The model searches for failures that might happen to the characters (e.g. becoming poisoned, dying), and when one is found, searches for a plan to avoid or negate that unacceptable state. The system uses representations of expected reader domain knowledge to identify what failures might occur. This knowledge is represented in the form of scripts. Scripts are conceptual frameworks, typically used to describe common situations. While scripts generally describe events, the script representation used by Dramatis also includes causal links, as in planning. (Scripts are described in greater detail in Section 3.4.3.) Causal links connect events where the effects of one event achieve the preconditions of a later event. Dramatis generates escape plans by attempting to "cut" a causal link—that is, it attempts to negate the proposition described by the causal link so that the preconditions of an anticipated subsequent action in the script can no longer be satisfied, and thus not be performed. In addition to no longer being able to reach the next action, the potential failure is also avoided, as it is now impossible to continue in the reader's script and reach that failure. Dramatis models reader memory in order to keep track of what story elements are salient in a reader's mind. This salience measure will be used to determine which solution to link-cutting will be most easily found by the reader, and thus how likely the reader will perceive that solution to be. The memory model uses a spreading activation model that tracks the events, characters, and objects that have reader focus. Events and objects that were used recently in the story are more strongly activated than those that appeared further in the past. Events and objects are connected when they occur at the same time. Thus, characters that are present throughout the story tend to be strongly activated, while items that have not appeared since the beginning

of the story will have comparatively low activation levels. The activation of these items factors into the cost of the previously mentioned planning task. The cost of the plan is equivalent to the perceived likelihood of the plan's success, which I equate to the level of suspense at that point in the story. Dramatis continues reading the story, tracking previously created escape plans as it reads, replacing them when it is clear that they are no longer the highest quality plan, or no longer viable given the current story state.

Figure 3 shows the general algorithm for Dramatis, broken down largely according to its components. In the following sections, I will describe the input to Dramatis, with extra emphasis on how Dramatis reads the story that it is given. I then describe how Dramatis retrieves scripts representing reader knowledge. The subsequent section describes the MEI-P Situation Model, which serves as a model of reader salience. I then describe the process that Dramatis uses to generate escape plans, the cost of which represents the level of suspense at that point. Ultimately, Dramatis outputs an ordered sequence of suspense ratings, and the escape plans that generated those ratings.

3.4 Dramatis System

3.4.1 Input

The input to Dramatis is a *Script Library*, an *Operator Library*, and a story in the form of an ordered set of *Time-Slices*. The script library is a (possibly empty) set of plan networks, representing the domain and genre information that one would expect readers to possess. The operator library is a collection of STRIPS operators [25], representing the actions that are available for planning tasks. The target story is given to Dramatis in the form of discretized time-slices which describe the actions and side effects of the story.

The *Script Library* is an optional input to Dramatis. When given, it contains one

Let SL be a script library, OL be an operator library, and T be an ordered sequence of time-slices. Let NM represent narrative memory, MS be a model of reader salience, and EP be the escape plan generated in the previous iteration.

DRAMATIS(SL, OL, T):
    $NM \leftarrow \emptyset$
    $MS \leftarrow \emptyset$
    $EP \leftarrow \emptyset$
    $EscapePlans[] \leftarrow \emptyset$
    $Suspense[] \leftarrow \emptyset$
    For each time-slice $t \in T$:
        Read t into NM
        $Scr \leftarrow$ Retrieve-Script(NM, SL)
        $MS \leftarrow$ Update-Salience-Model(NM, Scr, MS)
        If t is the next action in EP:
            $cost \leftarrow$ Recalculate-Plan-Cost(EP, MS)
        Else:
            $Links[] \leftarrow$ Identify-Links-To-Break(Scr, NM)
            For each link $L \in Links$:
                $\langle plans[L], costs[L] \rangle \leftarrow$ Generate-Escape-Plan(L, MS, NM, OL)
            $cost \leftarrow \min(costs[L])$
            Let EP be the plan in $plans[]$ with minimum cost in $costs[]$
        $EscapePlans[t] \leftarrow EP$
        $Suspense[t] \leftarrow cost$
    Return $\langle EscapePlans, Suspense \rangle$

Figure 3: Dramatis Algorithm

or more scripts that are used by Dramatis to predict what might go wrong for the characters of the story. When not given, Dramatis has no background knowledge for predicting failures unless provided with a character's plan by a time-slice.

The *Operator Library* is a set of uninstantiated STRIPS operators. Any operator used in a script in the Script Library or in the story time-slices must be contained within the operator library.

Dramatis reads in a story in the form of *Time-Slices*, which form a discretized version of the story. Each Time-Slice represents a single action in the story.

3.4.2 Time-Slice Interpretation

Dramatis reads the story one Time-Slice at a time. Time-Slices are discretized symbolic representations of the actions and states of the story. The time-slices are intended to be equivalent to a natural language story.

Definition 7 (Time-Slice): A *time-slice* is a tuple $\langle o, D, E, l, C, s, B^s] \rangle$ where o is an instantiated STRIPS operator, D is a set of propositions representing character dialogue, E is a set of propositions representing the effects of the time-slice on the world state in the story, l is a symbol representing the scene's location, C is a set of symbols representing the characters in the scene, s is an optional script, and B^s is a set of variable bindings for the operators in s.

Each time-slice describes exactly one action in the story. Time-Slices contain the instantiated STRIPS operator representing the action that occurred in the story. They also contain two sets of propositions, one representing the content of character dialogue, and the other representing the effects of the time-slice. The effects of the time-slice may be different from the expected effects of the operator in the time-slice. For example, the effects could include the fact that new characters are in the scene, or that an operator did not achieve all of its expected effects. Semantically, the

```
Action:  deliver-food (Waitress, vodkaMartini, James_Bond)
Location:  Casino Royale
Characters:  Waitress, James_Bond
Dialogue:  none
Effects:  at(James_Bond pokerTable) place(pokerTable)
Pacing:  medium
```

Figure 4: Example Time-Slice

dialogue propositions represent what a reader would infer from the dialogue in the scene. Algorithmically, these propositions are treated the same as the other effects propositions, because I assume that readers believe everything conveyed by dialogue. The Time-Slice contains a symbol representing the location of the action, and a set of symbols representing the characters in the scene. Time-Slices may optionally contain a character's plan in the form of a script. If so, the time-slice also contains a set of bindings from the operator variables in the script to a set of terminals in the story, representing characters or objects.

Dramatis has a narrative memory, which it uses to track the state of the story world using information provided from time-slices. The world state is updated in a two-stage process. In the first stage, the anticipated effects of the action operator are added to (or removed from) the story state. In the second stage, the propositions representing dialogue and side effects are added to the world state. Thus, when information from the operator conflicts with dialogue or side-effects, the operator effects are negated by the dialogue and side-effects. The narrative memory does not track the character and location information from the time-slices, but this information is used by MEI-P, my extension of the MEI Situation Model. When a character's plan is included in the time-slice, it is added to the script library.

Figure 4 shows a time-slice that was used during evaluation. In this time-slice, the operator in use is `deliver-food` (shown in Figure 5) with the parameters `Waitress`, `vodkaMartini` and `James_Bond`. The location of the time-slice is `Casino Royale`,

```
op:  deliver-food (?waiter ?food ?customer)
    con:  person(?waiter) person(?customer) edible(?food) (neq
        ?waiter ?customer)
    prec:  has(?waiter ?food) ordered(?customer ?food)
        waiter(?waiter)
    add:  has(?customer ?food)
    del:  has(?waiter ?food) ordered(?customer ?food)
```

Figure 5: Example Planning Operator

and `James_Bond` and `Waitress` are identified as characters. There is no dialogue information in this time-slice. The effects of the instantiated `deliver-food` operator will be added to the story state, as will the effects listed in the time-slice. The effects field adds information which has just appeared in the scene in the natural language version of the story—specifically, that James Bond is sitting at the poker table. Finally, while some time-slices contain character plans, this one does not.

3.4.3 Script Identification

Once Dramatis has read in more than one time-slice, it searches the script library to determine if any scripts match the observed events. These scripts represent domain and genre knowledge on the part of the computational reader. Dramatis needs to be bootstrapped with the knowledge typically possessed by human readers, such as background knowledge about common situations and past stories that they have read.

Broadly, scripts are conceptual frameworks used by people to navigate common situations (e.g. going to a restaurant) [68]. Typically, people learn these scripts through personal experience. However, it is also possible for people to acquire scripts through second-hand tellings of others' personal experiences. For example, people can acquire scripts about car accidents without ever having been in one themselves. Schank provided one of the first computational representations of scripts. In the 'Schankian' model, scripts represent causal chains, where earlier events in the script allow for the events occurring at the end of the script. For example, food must be

ordered from the server before the food arrives or paying the bill. Despite the length of a script, people typically need only observe a small number of events in order to recognize that a script is relevant. This ability to discern a script from a small number of observations allows us to fill in the gaps when events are left out of stories [68]. Thus, when we hear stories about going to restaurants, we are able to fill in that a person gave their order to a waiter, whether or not the storyteller provides us with that detail. Scripts, therefore, are a significant tool in how people understand stories [23].

In addition to having scripts for basic situations like the restaurant described above, I argue that people have scripts for common story situations, such as bank robberies or spies being poisoned. If people are able to acquire scripts second-hand about car accidents then they can just as easily acquire scripts representing genre knowledge (e.g. horror films, spy movies). Just as people learn how to interact in a restaurant through personal experience, audiences learn these common story tropes through repeated exposure. Because people have knowledge of the events and causality of typical story situations, collected through personal experience, we can use scripts in Dramatis to represent genre or domain knowledge on the part of the reader. In the Dramatis model, scripts serve to give the model an understanding of what types of events occur in these kinds of common story situations, as well as what negative consequences they might lead to for the protagonist. For example, Figure 6 shows a fragment of a script wherein a spy is poisoned and eventually dies. Scripts in Dramatis come from one of two sources. Most scripts come from the script library which is given as input to the Dramatis system. These scripts represent reader genre knowledge and indicate what sequences of events can be expected to lead to some negative consequence. The second source of scripts are time-slices, which may include a character's plan in the form of a script. When the audience is made aware of a character's plan (usually, the plan of the antagonist), then we can treat that plan

as a sequence of events leading to some negative outcome for the protagonist.

Plan networks are one of several updates to the Schankian script model [58]. Plan networks have a directed graph structure. In these plan networks, nodes represent individual events, while directed edges connect a node to those events that could possibly succeed the original node temporally. Thus, by following these temporal links between events, we can traverse the graph structure to generate a valid sequence of events in a given domain. The plan network, therefore, has a set of valid traversals, each of which represents a possible sequence of events for its particular context. This collection of sequences allows for more freedom as compared to Schankian scripts, which typically have a linear structure, and therefore have a less detailed understanding of the possible events in a scene.

Dramatis scripts are more similar to Orkin's plan networks [58] than to the scripts described by Schank and Abelson [68]. Dramatis uses the plan network representation because it allows scripts to be represented as sets of possible sequences of discrete events. Additionally, Schankian scripts typically take the point-of-view of a single person in a scene (e.g. the restaurant customer). In the context of stories, it makes more sense to be able to shift between the points-of-view of multiple actors. Orkin's plan networks assume interaction between multiple actors.

Dramatis uses a script structure based on Orkin's plan networks, which I define as follows:

Definition 8 (Script): A *script* is a graph $G = \langle V, T, C, B \rangle$ where V is the set of script nodes, $T \subseteq V \times V$ is the set of temporal links, C is the set of causal links, and B is a set of co-designation constraints for the variables within the operators within the script nodes V.

A script is a graph containing the script nodes, representing events, and the causal and temporal links between the nodes. Additionally, a set of co-designation constraints indicates when a set of variables in the STRIPS operators (from the script

nodes) should be bound to the same terminal when applied to specific story information. For example, a co-designation constraint would indicate that some set of script nodes represent actions that should all be performed by the same character. Neither the script nor the co-designation constraint indicates which character this should be. It only indicates that binding any one of these variables to a symbol (e.g. a character) will bind every other variable in the co-designation constraint to the same symbol.

Definition 9 (Script Node): A *script node* is a tuple $\langle o, x \rangle$ where o is an uninstantiated STRIPS [25] operator, and x is a boolean flag that represents whether the node has been defined as an Exit Node.

Script nodes are pointers to STRIPS [25] operators in the operator library. Uninstantiated operators use variables to represent the characters or objects that are parameters to the operator. Dramatis will instantiate the operators, binding the variables to the operator parameters specified in the time-slices, when it attempts to identify a script that is relevant the the story being read. This binding procedure uses a simple unification process, matching the operator name to observed actions, and making sure that the script's co-designation constraints are not violated as the bindings are constructed. If the constraints would be violated, the binding process backtracks to other nodes in the script. Exit Nodes indicate that the traversal of a script could end at a given node. In the specific context of Dramatis, Exit Nodes have negative consequences for the protagonist of the story, such as falling unconscious or dying. It does not matter which of the effects of the Exit Node's operator is the negative consequence. Dramatis will simply try to prevent this action from occur by preventing its preconditions from being satisfied, thus preventing all of the effects from becoming true in the world.

Definition 10 (Temporal Link): A *temporal link* is a tuple $\langle a_i, a_j \rangle$, where a_i and a_j are events. The temporal link means a_i immediately precedes a_j.

Definition 11 (Causal Link): A *causal link* is a tuple $\langle a_i, a_j, p \rangle$, where a_i and a_j are actions, and p is a proposition, such that p is an effect of a_i and a precondition of a_j.

Script nodes are connected by two different types of links. *Temporal Links* are directed edges that indicate that the sink event may directly follow temporally from the event represented by the source node. These links are equivalent to the directed edges that existed in Orkin's plan network structure.

My script representation adds an additional set of edges that signify *Causal Links*, which are not part of the Schankian script representation nor Orkin's plan networks. Because each script node contains a STRIPS operator with preconditions and effects, I can link events that achieve predicates as effects that are necessary preconditions for later events. This causal information will be used later in the escape plan generation phase. Causal links are also directed edges that indicate that an effect of the STRIPS operator in the source node achieves one or more preconditions of the STRIPS operator in the sink node. This definition of a causal link is functionally the same as Definition 6 in the context of planning.

In Figure 6, each node shows the operator that the script node represents. Note that each of the parameters are variables, not specific terminal values from a story. The black arrows represent temporal links, while the red dashed arrows show causal links. Exit nodes have red outlines. This diagram does not show the co-designation constraints. See Figure 7 for one of the full scripts used during Dramatis evaluation.

After reading in the time-slice, Dramatis conducts a linear search of every script in its script library to see which script is the best match for the story-so-far. Even when matches have been found in previous iterations, the search is repeated in order to be certain that a better match cannot been found. Matching scripts are determined by how many of the observed events are used in the script, whether the ordering of story events matches the expected temporal orderings of the script, and whether

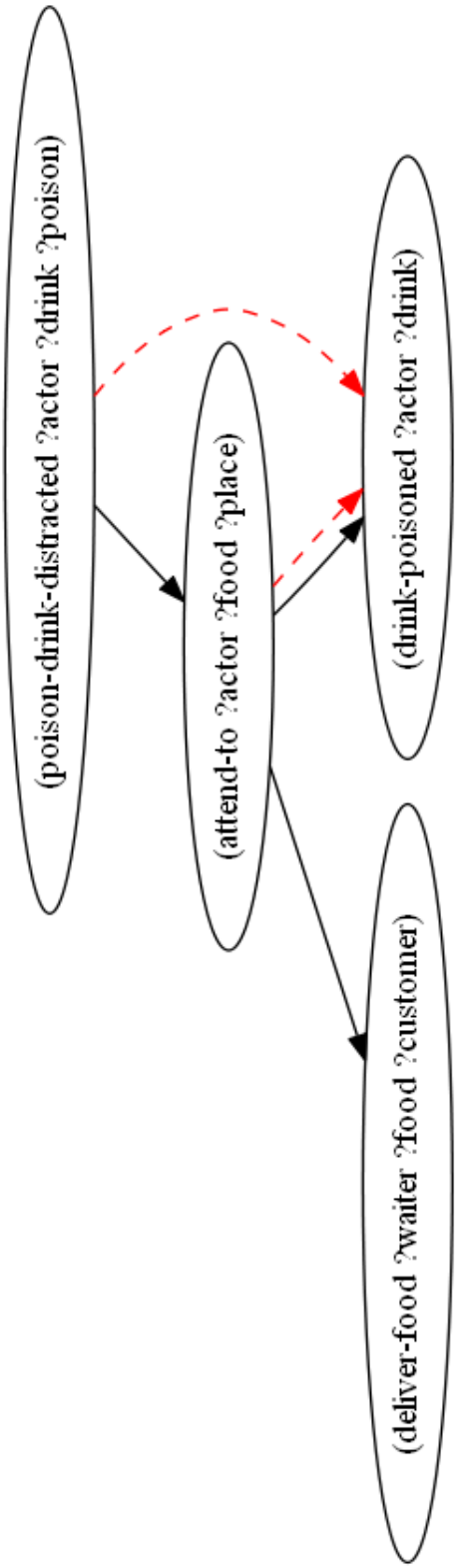

Figure 6: A fragment of an example script.

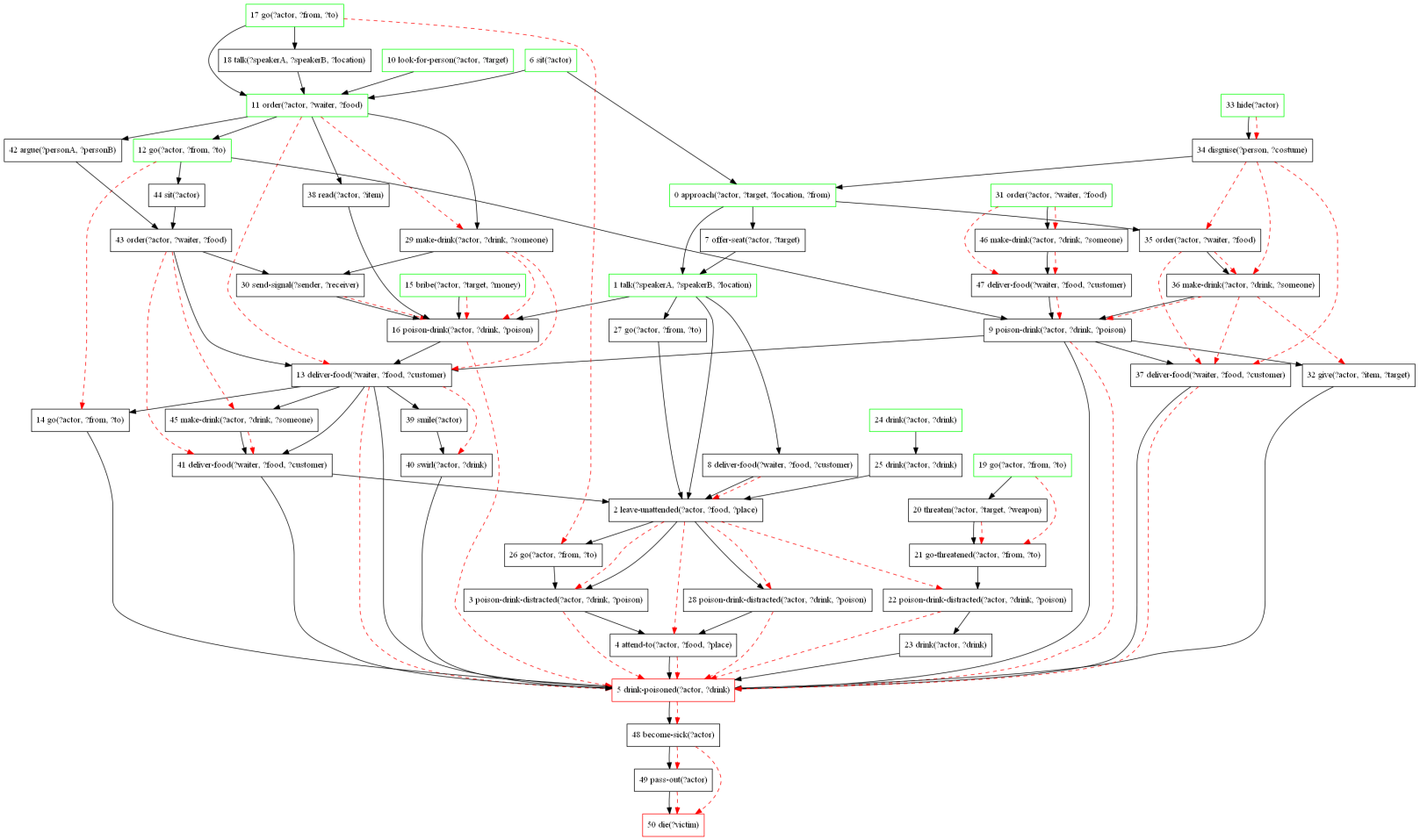

Figure 7: An example script used in Dramatis evaluation.

Dramatis can bind characters and objects in the story to script nodes while following the script's co-designation constraints. Finally, scripts get extra credit if they are actually character plans that were added to the library from a time-slice.

The score S for a script *scr* is equal to:

$$S_{scr} = E * c_E + O * c_O + P * c_P + B * (c_B + act_B) \quad (1)$$

where:

- E is equal to the number of events that have been observed in time-slices and are part of the script *scr*
- O is the number of pairs of sequentially observed events that are valid temporal orderings according to the script *scr*
- $P = 1$ if the script is a character's plan, otherwise $P = 0$.
- $B = 1$ if a valid set of bindings exists, otherwise $B = 0$.
- c_E, c_O, c_P, and c_B are constants. (As implemented, $c_E = 1$, $c_O = 1$, $c_P = 3$, and $c_B = 3$.)
- The term act_B is the activation of all elements of the bindings according to the memory model. When more than one valid set of bindings exists, S_{scr} uses the maximum value of act_B across all valid bindings.

  $$act_B = \max_{b \in B}(\sum_{e \in b} activation(e)) \quad (2)$$

  where B is the set of all valid bindings, b is one set of valid bindings within B, and e is an element of the story and the memory model (e.g. a character, event, or item).

Equation 1 is one of several approaches that would be successful at retrieving scripts. In particular, with a larger number of scripts, a more efficient search process,

where scripts were indexed in advance, would be useful for retrieval. The noteworthy part of this approach is the use of the memory model, which allows Dramatis to favor scripts that bind well with elements that are strongly activated in memory.

Dramatis retains the script with the greatest S_{scr} as a match, so long as $S_{scr} > 0$. In addition to retaining the script, it retains the best set of bindings. These bindings will be useful later in the escape planning process.

In addition to tracking all observed time-slices, the Dramatis narrative memory maintains a set of information relating the matching script to the story. Specifically, Dramatis maintains a list of script nodes that correspond to the observed events of the story. Additionally, Dramatis maintains a pointer to the current script location, which is the most recently observed node in the script. The script and the current script location are also used by the MEI Situation Model to predict future events.

As an example, suppose we have observed time-slices with the following 3 actions, in this order:

- `poison-drink-distracted(Valenka, vodkaMartini, inkPoison)`
- `attend-to(Waitress, vodkaMartini, bar)`
- `deliver-food(Waitress, vodkaMartini, James_Bond)`

The script in Figure 6 would score credit for containing each of those three actions. Additionally, because those events are in the same order as the script, the score would increase for matching the event ordering. The script would get binding credit if it were possible to bind the symbols in the time-slices with the operator variables in the script without violating the script's co-designation constraints.

While scripts are typically retrieved from the script library given as input, there is an additional source of scripts in Dramatis. As mentioned above, time-slices may contain characters' plans, such as when a villain gives a monologue detailing his plan to the hero. These character plans are simple scripts with a set of bindings to the

characters and items in the story. While scripts provide clues to what readers may expect to happen given their domain or genre knowledge, character plans are specific information provided by the story about how things will happen. Character plans receive a bonus when scoring scripts during script retrieval because they have been specifically referred to in the story.

3.4.4 MEI-P Situation Model

As Dramatis reads the story, it tracks the events and objects in the story using a memory model. This memory model is based on the MEI Situation Model component of Niehaus' INFER system [57]. I refer to this memory model as the Modified Event Indexing with Prediction (MEI-P) Model. Like MEI, MEI-P is a spreading activation network, where greater activation for an event or object indicates greater salience for that particular element. Narrative psychologists, such as Zwaan [81], have proposed spreading activation models in the past as a means of representing the memory and focus of readers. In MEI-P, story elements with higher activations are more easily retrieved for the purposes of identifying scripts or generating plans. Recall that MacLeod and Campbell [48] argue that the first thing that comes to mind is perceived by a reader to be the most likely thing to occur. Thus, a memory model based on salience can help a planner determine what story elements will be perceived by readers to be most likely. Therefore, I use node activation in the MEI-P Situation Model as a means of calculating operator costs in the planning process, allowing more easily retrieved story elements to have lower costs for planning.

In the remainder of this section, I will provide additional background about MEI, both by describing its history in Zwaan's Event Indexing model and by describing its general functionality in INFER. I will then detail how I use the MEI-P Situation Model in Dramatis. Finally, while describing the mechanics of MEI-P, I will also lay out the changes that I made to MEI in developing MEI-P. The largest of these

changes is the inclusion of predicted events and story elements in the situation model. These elements are taken from the script currently active in Dramatis. As a result of this change, I alter the equations used to calculate the activation of the edges and nodes in MEI-P.

Zwaan et al. [81] proposed the Event Indexing (EI) model as a means of representing a reader's situation model of a story—that is, the reader's mental model of the story as it is being read. In the EI model, a reader's situation model is updated with each new story event. Each event is described in terms of its protagonist, the time when it occurs, the space where it occurs, its causal relationships with other observed events, and its connections to the intentions of the characters of the story, particularly the protagonist. The situation model contains indices for each of these five categories (protagonist, temporality, spatiality, causality, and intentionality), which are updated when a new event indicates that one of the values has changed. Further, they propose an activational model wherein activation is tied to recency. When discontinuities are observed in one of the indices (e.g., a change of location), the activation of the previous value (the old location) decreases, while the new value becomes activated. When activation values decrease, a reader's ability to process the new information is slowed.

Niehaus developed the Modified Event Indexing (MEI) model as a computational representation and expansion of the EI model, as part of the INFER system [57]. Niehaus describes the 'modification' from the EI model as the formalization of the model for computational implementation. The purpose of INFER was to model narrative focus, using a spreading activation model between recently mentioned elements to other elements of the story. Further, INFER uses the MEI model to determine what inferences a reader would be able to make in the story, based on both story knowledge and whether the necessary story knowledge is activated in the reader's mind.

As a component of INFER, MEI is an incrementally constructed model of the story

as experienced by the reader as a means of determining the degree of reader focus. MEI is divided into two components: the *Situation Model* and the *Reader's Story*. The Situation Model is a semantic network used to compute the salience of story elements in the reader's mind. Elements of the story, based on the dimensions of space, time, causality, characters, objects, and intention (the same five dimensions tracked by Zwaan's EI model), are connected in this network based on relatedness. Elements are related when they are referenced at the same time in the story. The connections between the story elements represent how focus spreads between elements as the story is read. The Reader's Story represents the structure of the story in-progress from the point-of-view of the reader, using a formalism from Riedl and Young [66].

Niehaus defines two types of nodes in the MEI Situation Model network: event nodes and dimension nodes [57]. The below definitions are adapted from Niehaus [57], but modified to represent changes made for the purposes of Dramatis.

Definition 12 (MEI-P Event Node): An *event node* contains the operator indicated by the corresponding Time-Slice in the story.

Definition 13 (MEI-P Dimension Node): A *dimension node* is one of four sub-types of nodes: space, time, causal, or object. Space nodes contain pointers to a location specified by a Time-Slice. Time and causal nodes contain no additional information. Object nodes contain terminals or predicates.

Niehaus includes *intention nodes* as a fifth type of dimension node. Intention nodes contain a character and a predicate, which represent a character with a goal. Because Dramatis does not use information about character goals, I do not include this type of dimension node. There is no indication that removing intention nodes from the situation model has any significant effect on its performance representing reader memory.

Niehaus defines the Situation Model itself as follows:

Definition 14 (Situation Model): A *situation model* is a weighted bipartite graph $G = \langle V^G, E^G, w^G \rangle$, where V^G is the tuple of nodes $\langle X^G, Y^G \rangle$, X^G is the set of event nodes, Y^G is the set of dimension nodes, and X^G and Y^G are the partite sets of V^G, $E^G \subseteq X^G \times Y^G$ is the set of edges, and $w^G : E^G \to \mathbb{R}$ is a weighting function over the edges.

Thus, the MEI Situation Model is a bipartite graph between the event nodes and the dimension nodes. The latter category describes relevant details of the events: specifically, their locations, time, causal relatedness to other events, characters, objects, and the preconditions and effects of the events. For example, a dimension node representing a character will be directly connected to each event in which the character is a participant. A dimension node representing a predicate, such as `poisoned(James_Bond)`, will be connected to any event where it is a precondition or effect.

Dramatis expands only on the Situation Model, although a form of the Reader's Story is tracked by the narrative memory in Dramatis. This memory maintains the set of time-slices that have been read, along with the state of the story world. Ultimately, the value of MEI for Dramatis comes in the form of the spreading activation Situation Model, which allows Dramatis to estimate the salience of particular story elements in the reader's mind. Because we know that readers perceive their first thoughts to be most likely [48], we use this model of reader salience to determine what story elements will be more easily retrieved by the reader. Specifically, Dramatis uses these values as part of the operator costs in the process of generating escape plans.

Dramatis expands the MEI Situation Model to include a set of nodes and edges that have not yet been observed but are expected to occur. These nodes come from the current active script. Dramatis selects the script that best matches what it has seen so far, and it tracks which nodes in the script have been observed. This script is used to infer what events are likely to occur next. Dramatis adds nodes based on the

information in this script to the situation model, because it is likely that the script and its elements would have some level of salience in memory, just like the observed elements of the story. Thus, the expanded situation model, which I refer to as MEI-P (Modified Event Indexing with Prediction), includes not only observed events in memory, but also the contents of the script that is active in memory. Dramatis assumes that the activation of predicted script events and their related elements decreases as we go further into the future of the script. Just as observed elements lose activation as their distance from the current event grows (unless the elements are re-activated by being mentioned again), the salience of future elements decreases as their distance from the current event increases. Because some nodes are observed and some predicted, I extend the definition of a MEI-P situation model as follows:

- Let X^G be the tuple of event nodes $\langle X^{G,obs}, X^{G,pred} \rangle$ where $X^{G,obs}$ is the set of event nodes derived from observed time-slices, and $X^{G,pred}$ is the set of event nodes predicted using a script.

- Let Y^G be the tuple of dimension nodes $\langle Y^{G,obs}, Y^{G,pred} \rangle$ where $Y^{G,obs}$ is the set of dimension nodes derived from observed time-slices, and $Y^{G,pred}$ is the set of dimension nodes predicted using a script, containing only object nodes.

- Let an *observed node* describe any node in the set $X^{G,obs} \cup Y^{G,obs}$, and let a *predicted node* describe any node in the set $X^{G,pred} \cup Y^{G,pred}$.

- Let an *observed edge* describe any edge $e : (x \leftrightarrow y) \in E^G$ where $x \in X^{G,obs}$ and $y \in Y^{G,obs}$. Let a *predicted edge* describe any edge $e : (x \leftrightarrow y) \in E^G$ where $x \in X^{G,pred}$ or $y \in Y^{G,pred}$. Let $E^{G,obs}$ refer to the set of observed edges in G, and $E^{G,pred}$ refer to the set of predicted edges in G. Note that $E^{G,obs}$ and $E^{G,pred}$ are mutually exclusive.

The memory model is constructed one event at a time, as the story is being

read. When reading in a new event from the story from time-slice i, we create a new situation model N_i, by doing the following:

1. Create a new event node e_i containing the observed operator op_i, and add it to $X^{N,obs}$.

2. Create a new space node d_i^s containing the location from the time-slice, a new time node d_i^t, and a new causal node d_i^c, adding each to $Y^{N,obs}$.

3. For each predicate p_j in the preconditions and effects of op_i, create a new object node $d_{i,j}^o$ and add it to $Y^{N,obs}$.

4. For each predicate p_j in the dialogue and side-effects of the time-slice, create a new object node $d_{i,j}^o$ and add it to $Y^{N,obs}$.

5. For each term t_k in op_i, create a new object node $d_{i,k}^o$ and add it to $Y^{N,obs}$.

6. Create an edge between e_i and all nodes in Y^N with weight equal to 1.0, and add it to E^N.

Figure 8 shows an example of how we convert from a single time-slice to the new model N for that particular event. The event node is the rectangle in the center of the diagram. The dimension nodes created for causality, space, and time for this event are shown as parallelograms. Finally, object nodes are shown as ovals in the diagram. We create object nodes for the terms of the event: `Waitress`, `JamesBond`, and `vodkaMartini`. We also create object nodes for the preconditions of the action in the time-slice – `has(Waitress vodkaMartini)` and `ordered(JamesBond vodkaMartini)` – as well as the effects – `has(JamesBond vodkaMartini)`, ¬`has(Waitress vodkaMartini)`, and ¬`ordered(JamesBond vodkaMartini)`. If there were dialogue or side-effects in this time-slice, we would add object nodes for the predicates in those categories as well. The diagram does not show edge weights, but in this case, each edge would have a weight of 1.0 because these are all new nodes.

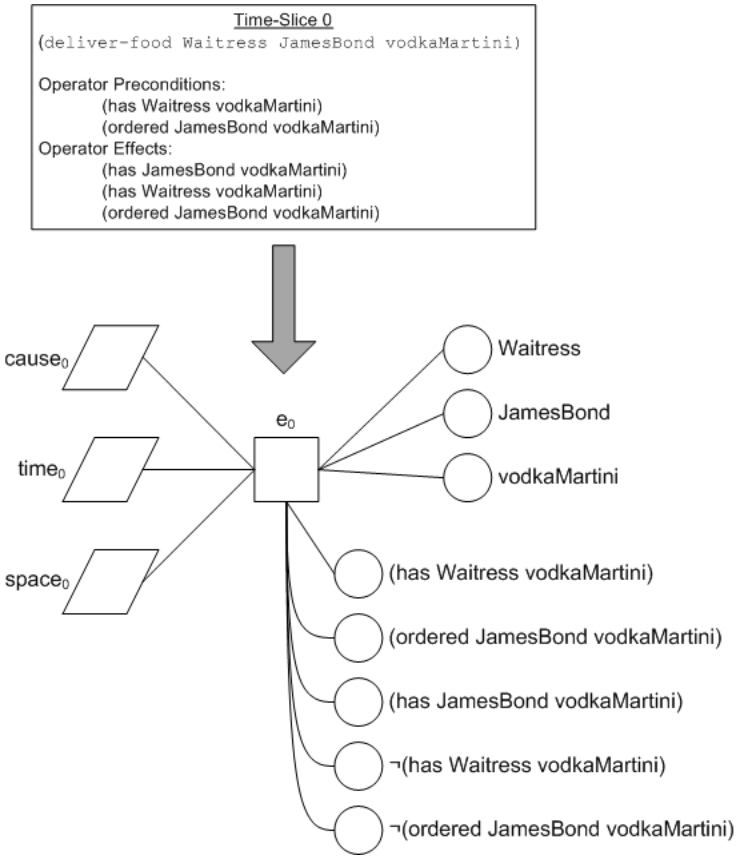

Figure 8: Example transition from time-slice to new MEI-P Situation Model

The next step is to incorporate the new model N_i into the existing situation model G. In the initial case, G is empty, gaining nodes only after Dramatis reads the events of the story. Figure 9 shows an expanded MEI-P Situation Model with multiple events. We create the full model according to the following process:

Let the full memory model after time-slice i be G_i, and the new model for the subsequent time-slice $i+1$ be N_{i+1}. The update function, $\uplus$, is defined as:

$$G_{i+1} = G_i \uplus N_{i+1} = \langle V_{i+1}^G, E_{i+1}^G, w_{i+1}^G \rangle \tag{3}$$

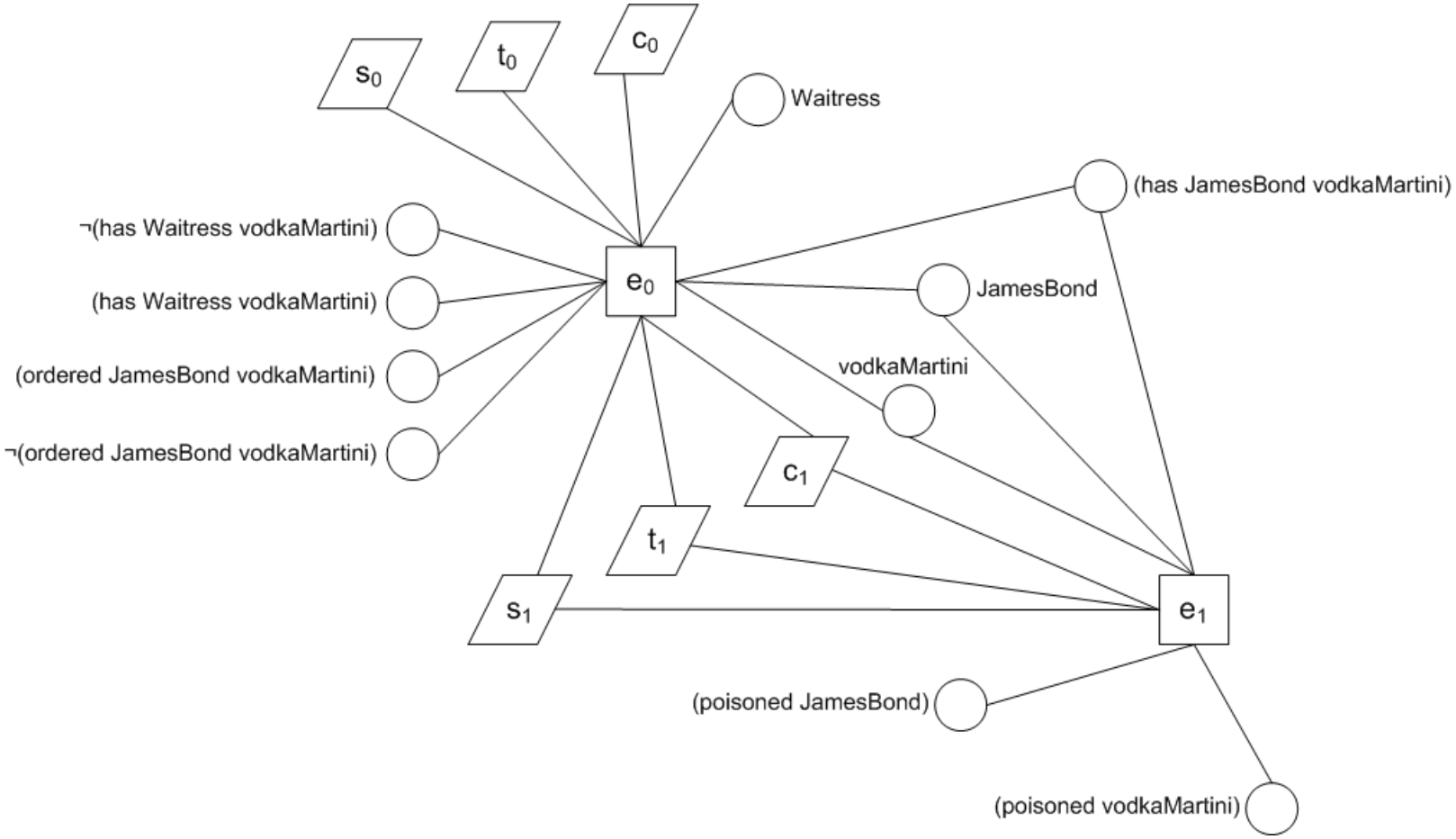

Figure 9: Example MEI-P Situation Model with multiple events

The nodes and edges of G_i and N_{i+1} are united into a single graph G_{i+1}:

$$X_{i+1}^{G,obs} = X_i^{G,obs} \cup X_{i+1}^{N,obs} \tag{4}$$

$$X_{i+1}^{G,pred} = X_i^{G,pred} - X_{i+1}^{N,obs} \tag{5}$$

$$Y_{i+1}^{G,obs} = Y_i^{G,obs} \cup Y_{i+1}^{N,obs} \tag{6}$$

$$Y_{i+1}^{G,pred} = Y_i^{G,pred} - Y_{i+1}^{N,obs} \tag{7}$$

$$E_{i+1}^{G} = E_i^{G} \cup E_i^{N} \tag{8}$$

All nodes in N were observed nodes, thus they are added to the set of observed nodes in G. When previously predicted nodes appear as part of N, they are added to the set of observed nodes and removed from the set of predicted nodes in G_{i+1}. Observed event nodes are considered to be the same as predicted event nodes if they refer to the same instantiated operator. In general, object nodes are considered to be the same if they represent the same predicate or terminal, or if they are co-designated according to the script.

The set of predicted nodes needs further updating, beyond the above sets of equations. To create the predicted nodes for the current situation model:

1. Let S be the current active script, if there is one. (If there is currently no active script, end this process.) Let $c \in S$ be the script node that was most recently observed in the story. Let V be the set of all nodes in S that have been observed in the story.

2. Using the bindings stored by Dramatis, instantiate the operators in all nodes of S. If no bindings exist for a variable, combine all references to variables that are co-designated in the script to a single variable.

3. Let D be the set $(S - V)$

4. For every script node $s \in D$, define d_s as the minimum distance from c to s, using any branch of the script. If no path exists from c to s, remove s from D.

5. Let P_{i+1} be a new situation model.

6. For every script node $n \in D$:

   (a) Create an event node e_n, assign it the predicted distance d_n, and add it to $X_{P,pred}$.

   (b) For each predicate p_j in the preconditions and effects of op_n, create a new object node $d^o_{n,j}$ and add it to $Y^{P,pred}$.

   (c) For each term t_k in op_n, create a new object node $d^o_{n,k}$ and add it to $Y^{P,pred}$.

   (d) Create an edge between e_n and all nodes in Y^P with weight to be assigned in a later step, and add it to E^P.

The above steps duplicate much of the creation of N, with two exceptions. First, each event node e_n is assigned a predicted distance between itself and the most recent observed event. We will use this distance to set the edge weight later. Second, we skip the process of generating space and time nodes, as scripts do not contain that kind of information. We can now update G_{i+1} accordingly:

$$X^{G,pred}_{i+1} = X^{G,pred}_i \cup (X^{P,obs}_{i+1} - X^{G,obs}_{i+1}) \tag{9}$$

$$Y^{G,pred}_{i+1} = Y^{G,pred}_i \cup (Y^{P,obs}_{i+1} - Y^{G,obs}_{i+1}) \tag{10}$$

$$E^{G,pred}_{i+1} = E^P_{i+1} \tag{11}$$

We have updated $X^{G,pred}_{i+1}$ and $Y^{G,pred}_{i+1}$ so that all nodes from P_{i+1} have been included, unless those nodes were already part of the observed set of G_{i+1}. Additionally, we have defined the set of predicted edges in G equal to only the set of predicted edges in P, thus removing any predicted edges from G which were not also in P. This allows us to remove predicted edges when our predicted script changes, or when predicted events are no longer reachable according to the script.

Figure 10 shows an example MEI-P Situation Model containing predicted nodes and edges. The observed nodes and edges are identical to those found in Figure 9.

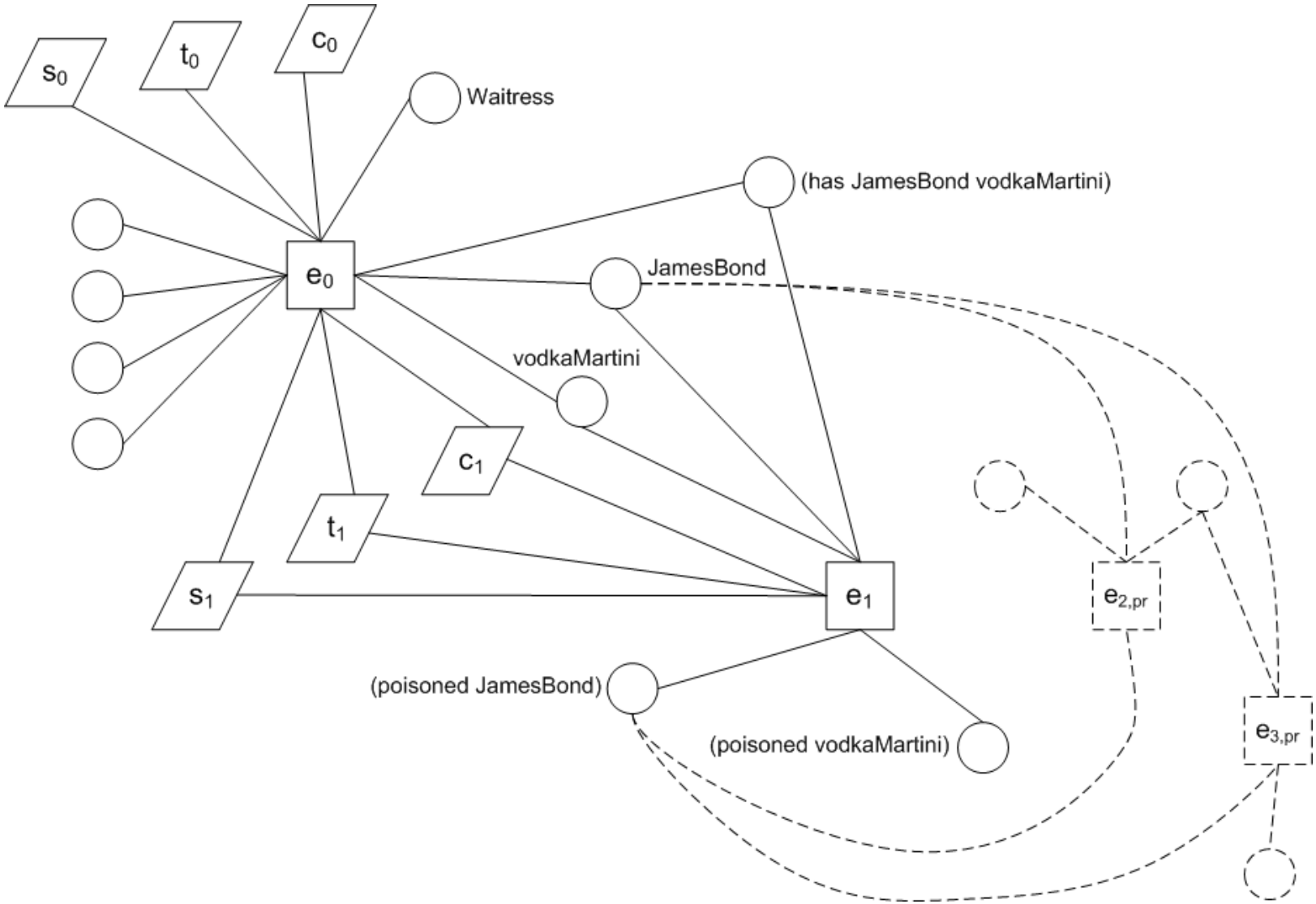

Figure 10: Example MEI-P Situation Model with predicted nodes and edges

Predicted edges are shown as dashed-lines, while predicted nodes have dashed outlines. This diagram shows a limited number of events predicted from the script, but in reality, the situation model would contain every event in the script that was reachable from the current location in the script. While edge weights are not shown, the weights on edges connecting to $e_{2,pr}$ would be greater than those connection to $e_{3,pr}$. This is because $e_{2,pr}$ comes earlier in the script. Also note that there are no causal, spatial, or time nodes for the predicted events. Dramatis only attaches these nodes to events that have been observed.

In an earlier step, we created dimension nodes for the new event, representing causality, space, and time. We now need to create edges to connect those nodes to previous events, where applicable.

- For all preconditions p of the operator in the event node e_i, create a new edge between the new causal node d_i^c and any prior event node that has p as an effect.
- If the last two time-slices were consecutive (i.e., no jump in time or flashback), create a new edge between the new time node d_i^t and the previous event node e_{i-1}.
- If the last two time-slices took place in the same location, create a new edge between the new spatial node d_i^s and the previous event node e_{i-1}.

As the story continues, the number of nodes in the MEI-P model is expected to grow at a logarithmic rate. While adding events causes new dimension nodes to be created each time, it is reasonable to expect that object nodes, representing characters, items, and predicates in the world state, will be reused. In the worst case, however, this growth would be linear. However, this would represent a story in which almost nothing followed from previous events. In either case, the number of edges is expected to grow linearly with the number of observed and predicted events.

Once the graph G_{i+1} has been formed, the next step is to update the edge weights w^G_{i+1} for all edges e, where e connects nodes u and v, and e_i is the new event node:

$$w^G_{i+1}(e) = \begin{cases} w^N_{i+1}(e) & e \in E^N_{i+1} \\ w^G_i(e) & e \in E^{G,obs}_{i+1} \text{ and } (u \in V^N_{i+1} \text{ or } v \in V^N_{i+1}) \\ S(D(w^G_i(e)), e, e_i) & \text{otherwise} \end{cases} \tag{12}$$

This process increases or maintains the weight of the edges closest to the new event node e_i, and discounts the weights of more distant edges. Equation 12 differs from its parallel in INFER. We add the middle condition, wherein a node retains its weight from the previous iteration if it has an edge connection to a node in V^N_{i+1}. I made this change because objects and characters that were frequently referred to in the story tended to lose activation despite their ubiquity in the story. By maintaining these weights at their previous level, we keep commonly referred-to objects highly activated in the situation model.

The function D is an exponential discounting function. Letting c_d be a constant:

$$D(x) = x * c_d \tag{13}$$

The function S gives defines the new weights for edges that are more distant from e_i. First, it uses the function SPD to determine the shortest-path-distance between e_i and the edge e. If e connects node x to node y, then the shortest-path-distance to e is the minimum shortest-path-distance between e_i and either x or y. Given this distance, a step function F discounts edge weight as e gets further from e_i. Let c_f be a constant:

$$F(x) = \begin{cases} 1/x & x < c_f \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Additionally, S uses an exponential term to determine how far in the future a predicted node is from e_i. Recall that when we created the predicted event nodes, we

stored a predicted distance with it. Further, recall that all edges connect an event node to a dimension node. Let e connect to the event node n, and if n is predicted, let its predicted distance be d_n. The function P defines the exponent in S, using the predicted distance:

$$P(e) = \begin{cases} d_n & e \in E_{i+1}^{G,pred} \\ 1 & otherwise \end{cases} \tag{15}$$

Finally, we can define the function S. Let v be a node adjacent to e along a shortest path to e_i with minimal degree, $D(v)$ be the degree of v, and c_s be a constant. The spreading weight function S is defined as:

$$S(x, e, e_i) = \frac{F(SPD(e, e_i))}{D(v)} * c_s + x^{P(e)} \tag{16}$$

As implemented, Dramatis sets the above constants as follows: $c_d = 0.5, c_f = 3$, and $c_s = 1$, which were the same values used by INFER.

Once the edge weights w^G have been calculated, it is then possible to determine the activation levels for each node in the memory model. Niehaus describes the activation calculation as part of a Markovian steady state distribution [57]. However, he also provides a simplified method for iteratively spreading node weights according to link weights. Dramatis implements this simplified method, described by the algorithm in Figure 11, setting $c_a = 0.01$. Calculating the activation for the nodes in the memory model is a linear operation, relative to the number of nodes in the model.

As an additional step, prior to updating the memory model, Dramatis prunes any node whose activation is extremely low. If $activation(n) < 0.5 * c_a$, then we remove all edges connecting to node n. The node itself is kept in the model, just in case the object or predicate is presented in the story. I prune these nodes because human memory is limited. If I were to retain every node, even when the activation was this close to zero, it would be akin to saying that a reader never forgets or ceases to be aware of every story element. From a computational perspective, nodes with

Let G be an MEI-P Situation Model, and let $a_i(n)$ describe the activation of node n after iteration i. Let $w(e)$ describe the weight of an edge e. Let $E(n)$ denote the set of edges, such that $\forall e \in E$, node n is a terminus of e. Let c_a be a constant.

For all nodes $n \in G$, set $a_0(n) = 1.0$.
`stable` $\leftarrow$ **false**
`i` $\leftarrow 0$
While `stable` is **false**:
    Increment `i`
    `stable` $\leftarrow$ **true**
    For all nodes $n \in G$:

$$a_i(n) \leftarrow \frac{\sum_{e \in E(n)} w(e) * a_{i-1}(x_e)}{degree(n)} \text{ where } x_e \text{ is the other terminus of } e$$

    If $(|a_i(n) - a_{i-1}(n)|) > c_a$ **then** `stable` $\leftarrow$ **false**

Figure 11: Algorithm for calculating node activation

such low activation levels were observed to drastically lower the overall activation of the network. When not pruned, such nodes would cause the activation levels of highly connected nodes to decrease, simply by virtue of being connected to them. Thus, it was possible for very common elements (such as the node representing the protagonist) to have its activation decrease because it was related to several elements which appeared only once and not recently. By pruning, the influence of these nodes is removed from the salience model, and the activation of commonly-used elements is left unaffected.

3.4.5 Escape Planning

In the Dramatis model, suspense is relative to the perceived likelihood of escape for the protagonist from some negative outcome, as stated in my reformulation of Gerrig and Bernardo's definition of suspense. Dramatis generates escape plans in order to produce a means of avoiding or overcoming that negative outcome, where the cost of the plan is equivalent to the level of suspense. Thus, the escape plan is an ordered

list of actions that leads to the aversion of the negative outcome.

Recall the definition of a plan (Definition 2) from Chapter 2.

> **Definition 2 (Plan):** A *plan* is any sequence of actions $\pi = \langle a_1, \ldots, a_k \rangle$, where $k \geq 0$ and actions are instantiations of plan operators [33].

Given some negative consequence, indicated by either a script or a character's plan, Dramatis needs three things in order to successfully be able to plan. First, Dramatis needs a goal situation to target. Second, Dramatis requires a means of calculating perceived likelihood. Finally, Dramatis needs a planning algorithm to use. Using the processes detailed below, Dramatis will search for the most likely escape plan once it has a script or character plan indicating a negative outcome. Otherwise, Dramatis may only skip the planning step when it is tracking an escape plan generated in a previous iteration.

3.4.5.1 Planning Problem Definition

A planning problem requires a goal situation to target in the search, an initial world state, and a planning domain. Dramatis uses the causal links in its current script to identify a goal situation. The causal link must be reachable from the current script node on any branch leading to an exit node. For its initial state, Dramatis generally uses the current state of the story world, which it has been tracking. Under certain circumstances (described later in this section), the initial state is constructed by advancing ahead in the script. The planning domain consists of the operators in the operator library, as well as states based on the predicates used within the story. The following definitions are reprinted from Section 2.6.

> **Definition 3 (Planning Problem):** A *planning problem* is a tuple $P = \langle \Sigma, s_0, g \rangle$, where Σ is a planning domain, $s_0 \in S$ is the initial state, and g is a set of propositions (referred to as *goal propositions* that must be true

Let S be the current script, and let $c \in S$ be the most recently observed node in that script. Let the operation $a \rightarrow b$ indicate that S contains a valid temporal path from node a to node b.

For all causal links $L \in S$, where L links nodes i and j:
    For all nodes $x \in S$ where x is an exit node:
        If $i \rightarrow j \wedge c \rightarrow j \wedge (j \rightarrow x \vee j = x) \wedge$
        $(c \rightarrow i \vee i \rightarrow c \vee c = i)$:
            Then L is a candidate link.

Figure 12: Algorithm for finding links to cut

in a state for a state to represent a goal situation. For any state $s \in S$, if $g \subseteq s$, then s is a goal situation.

Definition 4 (Planning Domain): A *planning domain* is a tuple $\Sigma = \langle S, O, \gamma \rangle$, where S is a set of states, O is a set of operators, and γ is a map $(S \times O) \rightarrow S$ is a state-transition function.

Definition 5 (Operator): An *operator* is a tuple $o = \langle$ n, PARAMS, PREC, ADD, DEL $\rangle$, such that n is the unique name of the operator, PARAMS are the variable symbols that appear within o, PREC is a set of propositions representing the preconditions of o, ADD is the set of propositions being added to the world state as effects, and DEL is the set of propositions being removed from the world state as effects.

The purpose of the planning task is to avoid or overcome the negative consequence indicated by the script. Remember that scripts also include causal information about the contained events. By cutting the causal link, the potential failure is avoided, as it is now impossible to reach the negative consequence in the current script because the necessary preconditions can no longer be met. Thus, Dramatis has a set of goals to target in its planning–the negation of any state indicated by the causal links.

Figure 12 shows the algorithm for determining what links are candidates to be cut by the planner. Any link that is still a candidate at the end of this process will have the negation of its predicate used as a possible goal. Dramatis begins by using the bindings retained from the script matching search to replace variables in the causal links with terminals from the story. If a causal link cannot be completely bound, we do not use it as a potential goal. Additionally, Dramatis maintains a list of causal links that have been successfully negated in the observed story based on completed escape plans. These causal links are also removed from the candidate set. A causal link is a candidate for planning search from the current script node if each of the following are true:

- There exists a temporal path (involving one or more temporal links) from the causal link source to the causal link destination, AND
- There exists a temporal path from the current script node to the causal link destination, AND
- There exists a temporal path from the causal link destination to some node that is classified in the script as an Exit Node, or the causal link destination is itself an Exit Node, AND
- There exists a temporal path, in either direction, between the current node and causal link source node, or the current node is the causal link source.

Because of the need to check for temporal paths surrounding each causal link, the running time of this search can be expressed as $O(|T| \times |C|)$, where T is the set of temporal links and C is the set of causal links (see Definition 8).

Figure 13 shows a simplified script with temporal links (in black) and causal links (red dashed lines). Assume that node C (starred) is the current node and node X is an exit node. Given that, the candidate causal links are B $\rightarrow$ D, C $\rightarrow$ D, and D $\rightarrow$ X.

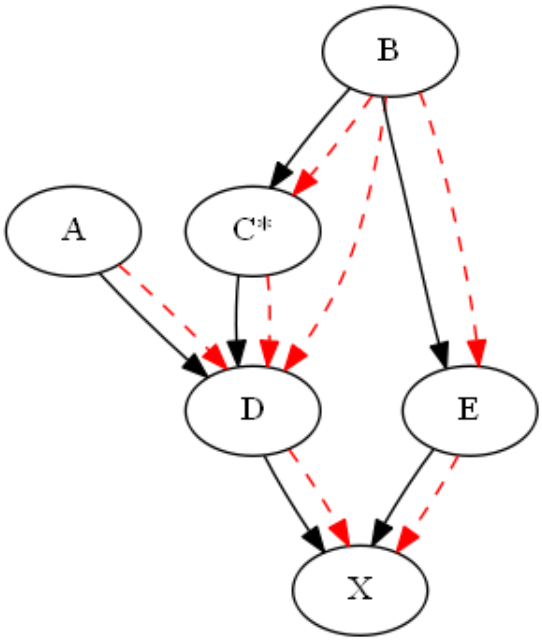

Figure 13: Example script fragment with temporal and causal links.

Thus, the goals for planning are the negations of whatever predicates are symbolized by those causal links.

We reject the causal links B → E and E → X because there is no temporal path between E and C. Similarly, we reject A → D because there is no path between current node C and node A. We also reject B → C as a candidate because it is entirely in the past.

After each time-slice, Dramatis updates its model of the current story world state. In many cases, this world state is the initial state for escape planning. However, it is possible that one of the goals indicated by the causal links is already true in the world. For example, a script may indicate that a goal is ¬`poisoned(James_Bond)`. However, if he has not yet been poisoned, then this state is already true in the world. For the purposes of planning, Dramatis follows the script to the node where `poisoned(James_Bond)` becomes true, updating the world state along the way. This updated world state is used as the initial state for the planning problem. Any actions used this way are considered *advance actions* and will be included in plan costs

and suspense calculation after planning. In the example above using Figure 13, this process might be necessary for the causal link D $\rightarrow$ X.

As with the case where a causal link cannot be bound, we do not plan in the case that the preconditions, effects, or the symbols within the advance actions cannot be bound. Attempting to do so would introduce unbound variables into the initial state of the planning problem, making actions inaccessible to the planner.

Restricting the planning task to this small set of causal links significantly reduces the planning task that Dramatis (and the reader) would otherwise face. Planning is a **PSPACE** problem, and the process for generating an escape plan requires multiple planning tasks. Generating the entire search space would be intractable for any domain of sufficient complexity, requiring $O(cb^n)$, where b is the branching factor of the search space, n is the depth of the search tree being generated, and c is the constant time required to generate a successor node in the search tree. Generating a single escape plan is $O(l(cb^n))$, where $l = |T| \times |C|$, because it is necessary to generate an escape plan for each link selected as a candidate using the algorithm in Figure 12. Although this increases the complexity because of the additional constant, I expect the average time to be reduced because Dramatis stops searching once it finds a solution, thereby reducing the necessary search depth, n. Additionally, the intelligent selection of planning goals informs the search, thereby reducing the amount of search required when compared to a less informed approach.

3.4.5.2 Operator Cost

Dramatis uses the MEI-P Situation Model to simulate activation of events, characters, and other story elements in reader memory. Our planning task attempts to use perceived likelihood as a measure of plan quality. We do this by assuming that the easiest thing for a reader to recollect from memory will be perceived to be the most likely thing to happen, based on research on memory retrieval by MacLeod and

Campbell [48]. That is, highly activated story elements are easier to retrieve. Thus, the costs of plan operators should be inversely correlated with the activation of the elements within the operator.

An operator's cost $Cost(op)$ is equal to:

$$Cost(op) = \begin{cases} \min(c_{max}, \frac{c_{op} + c_{prec} + c_{eff} + c_{term}}{A}), & \text{if } A > 0 \\ c_{max}, & \text{if } A = 0 \end{cases} \tag{17}$$

where c_{max} is a constant representing the maximum operator costs, and c_{op}, c_{prec}, c_{eff}, and c_{term} are also constants.

A is the weighted activation of the elements of the operator, which is calculated as:

$$A = O + P + E + T \tag{18}$$

where O is the activation of the instantiated operator, or any version of the operator, if the memory model does not contain the instantiated version, P is the mean activation of the operator preconditions, E is the mean activation of the operator effects, and T is the mean activation of all of the terms that are parameters to the operator.

$$O = \begin{cases} c_{op} * activation(op), & \text{if } activation(op) > 0 \\ c_{action} * activation(action), & \text{otherwise} \end{cases} \tag{19}$$

As an example, suppose we are trying to calculate the cost of the operator `deliver-food(Waitress vodkaMartini James_Bond)`. If MEI-P contains that full operator already, and its activation is non-zero, we apply that node's activation to the operator cost. Alternately, when its activation is zero, or MEI-P does not contain a node representing the full operator, we search for a node representing the `deliver-food` operator, regardless of its parameters. In that case, we apply the activation for that node instead.

$$P = c_{prec} * \frac{\sum_{p \in prec(op)} activation(p)}{|prec(op)|} \tag{20}$$

$$E = c_{eff} * \frac{\sum_{e \in effects(op)} activation(e)}{|effects(op)|} \tag{21}$$

The above two equations show the calculation of the mean activation for preconditions and effects of the operator. I use mean activation so that no operator is penalized for having a small number of preconditions and effects.

$$T = c_{term} * \frac{\sum_{t \in terms(op)} activation(t)}{|terms(op)|} \tag{22}$$

The final component of operator activation is the mean activation of the terms of the operator. Using the above example, T would be equal to the mean activations of `Waitress`, `vodkaMartini`, and `James_Bond`.

In my implementation of Dramatis, each of the constants in this section (c_{op}, c_{action}, c_{prec}, c_{eff}, and c_{term}) were set to 1. Retrieving the activation for a particular operator or proposition requires a linear time search for the corresponding node in the memory model and a constant time retrieval of the node's activation value. If no corresponding node exists in the model, the activation is considered to be zero.

In some domains, it may be worthwhile to consider other factors for operator costs, such as their objective likelihoods. For example, if a planning domain contained the operator `alien-abduction`, we would expect that to have a lower *a priori* likelihood than a common `go` operator, assuming no prior activation in the MEI-P Situation Model. In that case, there should be additional cost to those events which are, by default, significantly less likely. However, in all domains that were used for evaluating Dramatis, the operators were relatively common events (in the contexts of the respective domains) that did not need further modification to cost.

3.4.5.3 Heuristic Search Planner

Dramatis uses Bonet and Geffner's Heuristic Search Planner (HSP) [9, 10] to generate escape plans. HSP is a type of informed state-space search that uses a relaxed form of the planning problem to efficiently estimate the distance to a goal situation. The distance to the goal situation is the estimated sum of the action costs for reaching a goal situation. More specifically, the HSP search algorithm is the same as the A* algorithm, with the caveat that the heuristic is not admissible. Therefore, the planning solution found by HSP is not necessarily the optimal solution. The HSP heuristic estimates the distance to the goal situation by calculating the sum of the number of actions required to reach each predicate in the goal situation, while using a relaxed planning problem ignoring *delete lists* (the set DEL in Definition 5) in the STRIPS operator definitions. In addition, HSP allows for non-uniform action costs [41]. Dramatis uses the activation levels from the MEI-P Situation Model to define action costs. Because MEI gives the difficulty of retrieval, Dramatis can translate the activation level into perceived likelihood. This allows Dramatis to identify the escape plan that is not only the most likely to succeed, but more importantly, the most likely plan for the audience to retrieve.

The Heuristic Search Planner algorithm assumes the use of STRIPS operators that use *add lists* and *delete lists* (ADD and DEL in Definition 5) for effects, and that operators only have positive preconditions. A side effect of the HSP relaxation of the planning problem is that the goal must also be positive, since a negative goal could only be achieved via the delete list, which is ignored by the HSP heuristic. However, our link cutting technique for generating goal situations frequently leads to negative predicates as goals, such as a character not having a certain item. To overcome this issue, I have modified the algorithm so that it may peek at each operator's delete list to search for the (negated) predicate. That is, HSP will look at operator delete lists to see if they contain the goal situation. Any other predicate in the delete list is

ignored, as instructed by the HSP heuristic described above. This change does not break the heuristic, not does it affect any of the properties of the HSP algorithm.

HSP is one of several planners which would suffice for Dramatis. Ultimately, a planner in Dramatis simply needs to be able to search for a goal situation using operators with non-uniform costs, with a guarantee of finding the cheapest plan. Many planning algorithms that meet these properties could be substituted for HSP with no significant change to Dramatis.

During the actual planning process, HSP searches for a plan for each individual predicate proposed as a goal. Remember that while defining the planning problem for each individual goal, we also define an initial state for this problem. Additionally, recall from Section 3.4.5.1 that this initial state is not necessarily the same for each proposed goal. Some initial states are constructed from the current world state, while others are constructed by moving forward in the script to the source node for the causal link in question. Thus, because there is no guarantee that the initial state is the same for each proposed goal situation, each one needs to be investigated separately. Dramatis retains the escape plan that has the lowest combined cost between the plan generated by HSP and the cost of all advance actions, when they exist. Ultimately, this is a design issue rather than an algorithmic issue with the planner. If the initial states of the planning problems could be combined, it would be possible to turn several planning problems into a single problem. Regardless, the results of the processes would be identical.

As implemented in Dramatis, HSP will terminate planning after it has investigated 150 states without finding a solution to the planning problem. Dramatis interprets this situation as representing maximum suspense. This threshold is arbitrary, but in testing, I found that increasing this value did not change the results of any planning problems.

3.4.5.4 Tracking Escape Plans

Dramatis records the escape plan generated after the previous time-slice. When the subsequent time-slice matches the next expected event of the escape plan or the next expected advance action, Dramatis does not search for a new escape plan. Instead, Dramatis assumes that its escape plan has been proven correct. It re-calculates the costs of the remaining actions of the escape plan (and, if necessary, the remaining advance actions) according to the updated MEI-P Situation Model. The updated cost becomes the new suspense level at this point in the story.

When there are no steps remaining in the advancing actions and escape plan, the escape plan may be considered to be successfully completed. We consider the causal link to be successfully broken, and thus the plan complete, so long as the state of the world corresponds to the expected state after the link breaking. The only way this would not happen is if the new time-slice contains side-effects indicating that the expected effects do not match the current state of the world. When an escape plan is completed, the causal link is stored in a set of successfully broken links. From this point forward, Dramatis will not attempt to break causal links corresponding to the predicate represented by this causal link. If Dramatis did not store this information, the escape planner would consider it a valid causal link to consider breaking. However, because this link has already been escaped, the planner would mistakenly believe that there was no valid solution from the current state to a state where the link is broken a second time. This would be an incorrect interpretation of the world state.

If the newly observed event does not correspond to the stored escape plan, Dramatis rejects the previous escape plan and conducts the escape plan generation process described above. This may lead to the same escape plan, but at a different cost, as the world state and the memory model have changed. Alternately, this may be an opportunity to find a different escape plan.

3.4.6 Dramatis Output

Following each time-slice of the story, Dramatis outputs an *escape plan* (if one was found) and a *suspense rating*. The *escape plan* is a totally-ordered plan for avoiding a negative outcome for a character. The *suspense rating* is a positive number equal to the cost of the plan, which corresponds to the quality of the escape plan, to use the Gerrig and Bernardo's terminology. Thus, suspense is equivalent to the cost of the highest-quality escape plan. I refer to the set of suspense ratings over the course of a story as a *suspense curve*. The horizontal axis of this curve is story time, while the vertical axis indicates suspense rating. Figure 14 shows the suspense curve generated by Dramatis for a scene from *Casino Royale*, which was used as part of the evaluation of the model (see Chapter 5).

Let the following function define the suspense rating at a particular Time-Slice:

$$S(P, cost(P)) = \begin{cases} cost(P) & \text{if } P \text{ is a valid plan} \\ 0 & \text{if } P = \emptyset \text{ and any problem could not be bound} \\ suspense_{max} & \text{otherwise} \end{cases} \tag{23}$$

where P is the escape plan and $cost(P)$ represents the cost of that escape plan. If no escape plan was found, let $P = \emptyset$.

As described above, Dramatis equates the cost of the cheapest escape plan to the suspense level at that point in the story. Further, when no plan can be found, Dramatis defines the suspense level depending on the reason why the planning task failed. If any planning tasked failed because variables in the planning problem could not be bound, we define the suspense rating as 0. This planning failure typically indicates that the reader model believes that something bad will happen, but the specifics, such as what will happen or who it will happen to, cannot be reliably determined.

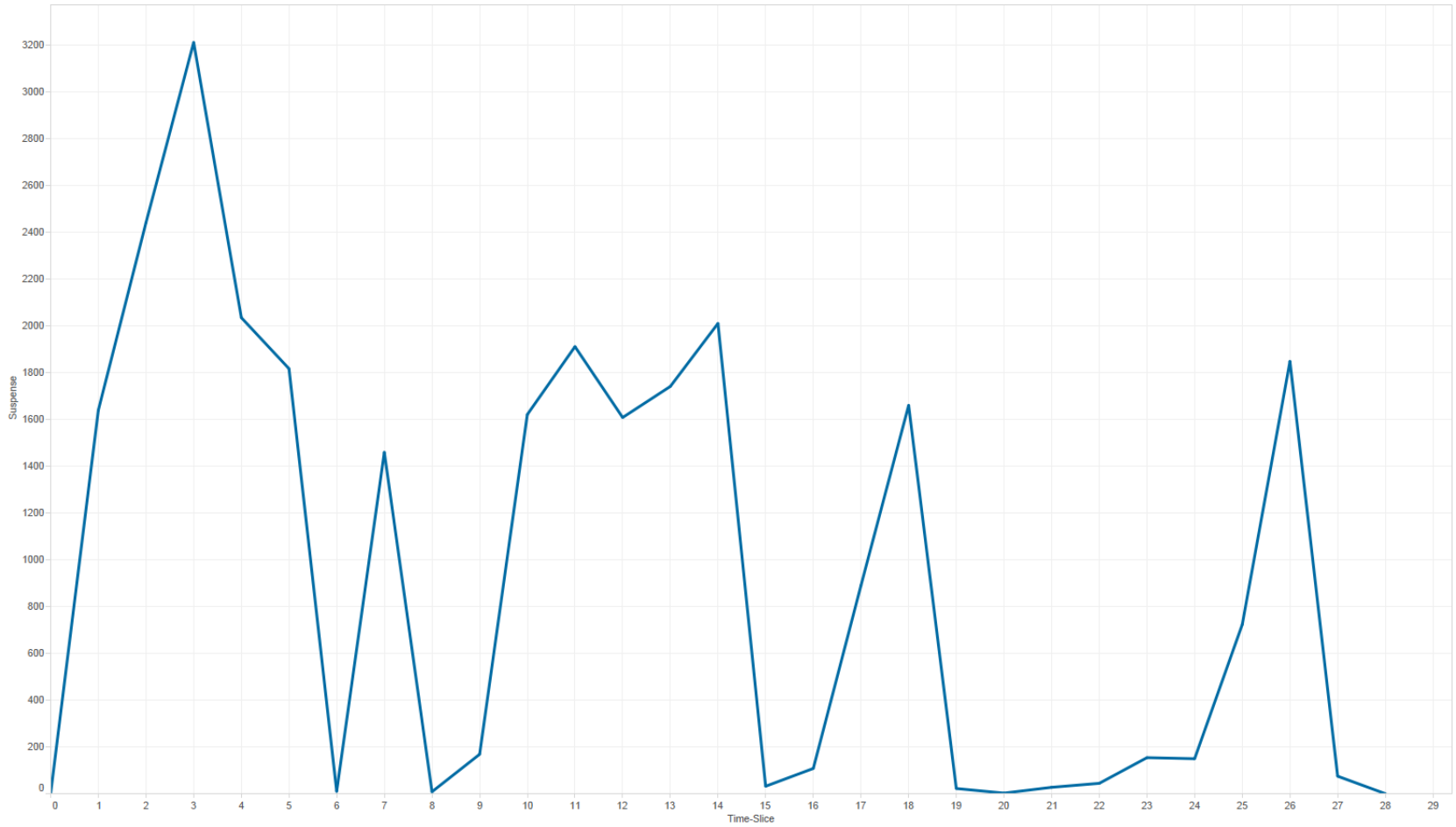

Figure 14: Example of a suspense curve produced by Dramatis

Thus the potential failure cannot be adequately defined, so I claim there is no suspense. When all planning failures are the results of the planner completing without finding a solution, we define this situation as maximum suspense. The situation is such that there is no viable plan within reach given the state of the world and the values within the memory model. Thus, we interpret the situation as a lack of quality plans, indicating a very high level of suspense.

3.5 Summary

In this chapter, I have introduced a computational human behavior model of suspense, based on a reinterpretation of Gerrig and Bernardo's definition of suspense. My reformulation expands on aspects of Gerrig and Bernardo's definition that are not sufficiently detailed for computation or computationally intractable as described. I make three observations:

1. Humans, being resource-bounded, do not generate the entire search space when problem-solving on behalf of the protagonist.

2. Paths through the search space may terminate for many reasons, several of which are consequences of the planning problem definition rather than the state of the story world.

3. Readers are incapable of perpetually considering the existence of suspense, and therefore do not repeatedly regenerate the search space.

As a consequence, my reinterpretation assumes that a reader can assume a negative consequence for the protagonist and generate the single plan, perceived to be most likely to succeed, that avoids or overcomes that consequence. My reinterpretation leads to the major components of Dramatis, including the MEI-P memory model, which provides information about perceived likelihood based upon salience

in memory, and the strategies for identifying goals for escape plans using reader domain knowledge, which helps reduce the complexity of an otherwise computationally intractable problem.

Because of the importance of the reader knowledge used by Dramatis, it is necessary to acquire and represent the knowledge of actual human readers. Without such knowledge, it is impossible to make a valid comparison between the output of Dramatis and the suspense reported by human readers. In the next chapter, I describe the process used to acquire this knowledge. In Chapter 5, I report on three evaluations of Dramatis. The first evaluation compares the output of Dramatis with self-reported suspense ratings from human readers. The other two evaluations are ablative tests, used to determine the value of the MEI-P memory model and the link-cutting strategy for goal identification, respectively.

CHAPTER IV

KNOWLEDGE ENGINEERING

Dramatis requires background knowledge, primarily in the form of scripts and planning operators, in order to understand stories and analyze suspense. Additionally, it is impossible to evaluate Dramatis against human readers if I am not certain that the knowledge possessed by Dramatis is comparable to the knowledge of human readers. Thus, a secondary objective for this work was a method for acquisition of the required knowledge from human readers in three domains for the purposes of experimentation and analysis. It was necessary to create a method that would allow me to collect typical reader genre knowledge while simultaneously limiting engineer bias while processing reader data. This chapter describes the processes used to acquire a corpus of natural language text and the conversion of that corpus into the knowledge structures required by Dramatis. There are significant challenges in both acquiring this type of knowledge and incorporating it into the system in ways that minimize engineer bias. In particular, the corpus is a set of narratives about specific scenarios, written as ordered lists of natural language sentences. Using methods adapted from qualitative research, I devised a means of converting this corpus into the necessary knowledge structures that mitigates engineer bias. This approach leads first to a set of planning operators, which are later combined into scripts based on the original natural language entries. This chapter describes the general process of converting a corpus into knowledge structures, such as operators and scripts, as well as the specific processes used in engineering knowledge for the domains used by Dramatis.

4.1 Background

4.1.1 Qualitative Methods

Coding is a qualitative research method used to elicit concepts, theories, or key phrases from natural language or visual data, such as journals, interview transcripts, videos, or other subjective data [67]. It is a common process in fields that work heavily with qualitative data, including learning sciences and HCI research. In many cases, coding is one step of a larger qualitative research process, such as grounded theory or thematic analysis. An individual *code* is a word or phrase that summarizes the key details of some aspect of the media being coded. In the case of interview data, individual codes may be applied at the paragraph level or multiple times per sentence, depending on the particular coding technique being used and the actual content of the data. The process of coding allows similarly themed data to be grouped together when codes are shared or overlapping. Coding is often an iterative process wherein codes are refined as researchers become more familiar with the data or attempt to create clear categories.

In the context of Dramatis, I generated a natural language corpus by asking participants to describe particular story scenarios. Corpora could be generated through other means as well, such as mining the web or using existing texts. Regardless of the source of the corpora, in the context of domain engineering, coding provides a technique for grouping entries that describe related actions, as well as identifying reasons why portions of the corpus may not be useful.

4.1.2 Knowledge Acquisition

For many AI systems that require background knowledge, such as scripts or plans (e.g. [23, 75, 52, 32]), these structures have been generated manually. However, manual generation leaves systems and evaluations prone to bias on the part of the domain engineers. As a result, other researchers have attempted to automate the

acquisition of procedural knowledge, such as scripts, from collections of natural language texts (known as *corpora*). Chambers and Jurafsky [17] learn "narrative event chains," structures that are similar to scripts but focus on a single protagonist, analyzing the Gigaword corpus to learn the major events of a chain. They proceed to use machine learning techniques to create a partial-ordering of these events. Fujiki et al [28] analyzed a corpus of Japanese newspaper texts in order to acquire scripts about murder cases. Kasch and Oates [39] extract script-like structures from a corpus of web documents pertaining to a particular subject. After narrowing the field of relevant documents, their system locates pairs of events based on argument co-reference. In each of these cases, researchers had access to a sizable corpus of text, such as newspaper articles about murder. In this research, it was first necessary to generate a corpus that would suit my needs.

Whereas the natural language approaches above attempt to recover structure directly from language, it is possible to use humans to generate specialized corpora. *Human computation* refers to systems that organize people to carry out computational processes, such as performing basic computational tasks that machines cannot typically perform effectively [44]. One particular form of human computation is *crowdsourcing*, where a computation task is distributed among a large pool of people, with the belief that the collective intelligence of the crowd of people is superior to that of a single person [44]. As human computation has grown, others have sought to use crowdsourcing techniques to acquire and aggregate procedural knowledge from large numbers of people rather than automating the learning process. Boujarwah et al [11] implemented a process for acquiring scripts from Amazon Mechanical Turk (AMT) workers, as well as using workers to classify the responses received in the initial script collection phase. Li et al [46] ask AMT workers to provide the typical events of a particular story, such as a bank robbery or a date. They further instructed the workers to keep their sentences simple and limit entries to a single verb. Having acquired their

corpora from the crowd, they then automate the learning of script-like plot graphs, which include relationships beyond temporal ordering. These approaches inform the instructions provided to my own participants, as well as the procedures used to check the validity of participant responses. Specifically, Boujarwah et al's iterative human processes minimized the level of machine processing. However, I believe the classification and quality control tasks that they assigned to AMT workers could be replaced with less-costly alternative methods, without diminishing the quality of the corpus. Further, while Li et al provide a valuable structure for collecting a corpus, I do not plan to automate the script-learning process.

4.2 Qualitative Knowledge Engineering Methodology

In this section, I introduce a qualitative knowledge engineering methodology as a means of converting a natural language corpus into a set of knowledge structures for an artificial intelligence system. I describe this methodology with the goal of being agnostic as to the source of the corpus and the particular representations of the knowledge structures. Given my goals for Dramatis, this procedure is specifically attuned to generating planning operators and scripts, and as a result, aims to extract actions or sequences of actions from the corpus. However, I believe that other similar knowledge structures could be generated using this methodology.

Given the broad applicability of this methodology, some decisions in this process are left to the researchers, who must decide what choices are most applicable to their goals. Further, researchers may find it useful to alter some of the details of this procedure in order to fit their system or representation. The remainder of this section describes the methodology broadly, while the following section details my implementation of the methodology with respect to Dramatis.

4.2.1 Creating a Corpus

This methodology presumes the existence of a natural language corpus that will be the source of knowledge for the system in question. The actual source of this corpus is not relevant to the procedure. Many of the procedures described above for acquiring knowledge begin with identifying corpora, each of which would be suitable for this methodology. Surveys, game traces, crowdsource responses, and web documents could all be applied to this approach. The best approach is dependent on the type of knowledge one hopes to encode. One expects to use different corpora for story knowledge as compared to commonsense reasoning, for example.

4.2.2 Coding the Corpus

The corpus is coded in a four-stage process adapted from qualitative methods processes used for parsing ethnographic data and interview transcripts, among other tasks. Each survey response or text sample in the corpus is treated as though it were an interview transcript. I refer to individual survey answers or sentences in the text sample as *entries*. In short, an entry should be the smallest unit of the corpus from which one hopes to extract actions.

The four phases of coding can be briefly described as follows:

1. Code the corpus by identifying actions, as well as potentially problematic entries within the corpus.

2. Combine actions and problems into broader categories, defining guidelines for what attributes indicate an entry belongs in a particular category

3. Multiple coders independently code a subset of the corpus, using the coding guidelines generated in the previous phase. Repeat this step until a sufficiently high level of inter-coder agreement is achieved.

4. A single coder from the previous stage codes the remainder of the corpus according to the same guidelines.

The first coding stage is based on a coding technique known as Initial Coding [67, 18]. Initial Coding is a "first cycle" coding method, wherein researchers develop tentative codes that will later be refined for analysis. This process also used aspects of In Vivo Coding [18, 69], which instructs researchers to create codes based on the actual words of the corpus. During this phase, a domain engineer codes each entry of the corpus. Entries that contain actions should be coded with the verb in the sentence. For example, the entry "The spy orders a drink" should be coded as the action "order-drink." If entries showed cause for concern for why they might not be valid for inclusion as an operator or script element, they should be coded with the reason why they might not work. Reasons for exclusion are specific to the coding task, and may not be applicable to all systems or domain engineering tasks. Potential reasons for exclusion could include ignoring a survey prompt, or referring to flashbacks (e.g., the entry "A flashback reveals the friend is a double agent" is marked as "flashback").

Once every entry had been coded, the next step is to perform a process known as Focused Coding [18]. Focused Coding is a common "second cycle" coding method that is used subsequent to Initial Coding. It is intended to identify patterns and categorize codes created during first cycle processes. Using this method, domain engineers combine the non-action codes into a taxonomy of codes that represent the space of possible rejection reasons. These codes form the basis of the coding guidelines that will be used in future phases. The exact number and breadth of these codes will be dependent on the domain or the corpus. Further, researchers should include a category for acceptable actions. However, researchers may find it useful to create broad taxonomies for action codes as well. For example, it may be useful to have separate codes for sentences representing single or multiple actions (e.g., "The spy

orders a drink" vs. "The spy sat down and ordered a drink"). For each code in the newly generated taxonomy, create guidelines that indicate when an entry should be given this code as opposed to other ones.

One could also consider the processes used in the first two phases as a form of Provisional Coding [67, 53], a means of establishing a set of codes prior to data analysis. Typically, these codes are based on prior related work and researcher hypotheses and expectations. The resulting set of codes can then be modified if data analysis reveals unexpected codes.

In the third stage, multiple coders independently code a subset of the survey responses using the codes and guidelines developed in the previous phase. A sufficiently high level of agreement and inter-rater reliability would indicate that a single coder could code the remainder of the corpus alone with a relatively low risk of error. Using multiple coders reduces the risk of error and increases the confidence in codes applied to survey responses. The corpus subset should represent approximately 20 percent of the full corpus. If multiple prompts were used in the creation of the corpus, be sure that the prompts are equally represented. When applying the action category (or sub-category, if appropriate) to a particular entry, researchers should specify the actions. This action coding mimics the In Vivo coding process used in the first phase of coding, as researchers should attempt to use the words used in survey responses. For example, one might code the sentence "The spy orders a drink" with the code `Action/order-drink`.

After the subset of the corpus has been coded, calculate inter-rater reliability. There are multiple useful inter-rater reliability metrics. In this chapter, I will commonly refer to Cohen's κ, but this particular metric is not a requirement of this process. For Cohen's κ, a score greater than 0.6 is typically considered "good," while values over 0.8 represent "excellent" agreement [43, 20, 26, 53]. Prior to beginning

this phase, researchers should decide what threshold of inter-rater reliability is sufficient for their task. Statisticians recognize that this threshold is arbitrary, ultimately dependent on the task in question and the importance of agreement [42, 67]. Referring to his own *alpha* measure of inter-rater reliability, Krippendorff suggests that a threshold of 0.667 could be applicable under certain circumstances [42].

While defining agreement in the case of the non-action codes is simple, it may be more challenging to define agreement for action codes. One might be content with simply classifying an entry as an action, or in the same action subset. For other tasks, agreement may be defined as using the same word to describe the action. Using In Vivo coding helps ensure that coders agree on the described action by applying the writer's own words.

After each iteration where inter-rater reliability was not high enough, coders should gather and discuss the codes and guidelines. Coders can revise codes or guidelines, or add new codes, in order to improve inter-coder agreement in the next iteration. Iterative processes such as this one are common in qualitative methods such as grounded theory [18]. These iterations continue until the inter-rater reliability threshold is met.

Once a sufficient level of inter-coder reliability is achieved, a single person can code the remaining entries from the corpus according to the most recent iteration of the coding guidelines. Additionally, the coder should resolve any remaining coding disagreements from the previous phase. This can be done through consensus agreement, or unilateral decision making. At the end of this process, every individual entry in the corpus is tagged as one of the following:

- An action, and what action is indicated. This may be further extended if coders used sub-categories.
- A candidate for rejection, along with the specific rejection code from the code taxonomy generated during Focused Coding.

4.2.3 Generating Knowledge Structures

The fully coded corpus can now be used to generate the knowledge structures needed. The specific processes for this conversion will depend on the representations being used (e.g. Schankian scripts vs. the script representation described in Chapter 3).

Broadly, each entry or sentence in every item of the corpus has now been coded as either an action or with a reason to reject or ignore the entry. These actions indicate the set of operators in the corpus. Depending on the system or representation, one may wish to simplify this set of actions by combining like actions. For example, entries coded as `Action/walk` and `Action/drive` could be combined into a more generic `go` operator, if the difference is not meaningful in the domain in question. If the coding guidelines did not include retaining information about who performed the actions or the causality of the actions, then these will need to be created by the domain engineers, perhaps by using the details in the corpus materials.

Given that each sentence of the corpus is now coded as an action, the larger items of the corpus (articles, survey responses, etc.) now contain sequences of items, which could be converted to scripts. As with operator generation, the exact details depend on the script formalism being used. Similarly, researchers will have to decide for themselves how to handle rejection codes in the middle of a text sample. Leaving details out could break the coherence of a script, but including bad information could likewise be harmful to the domain.

In the cases of planning operators, scripts, or any other knowledge structure, the formalism will define much of this process. Researchers should keep their formalisms in mind when going through the Initial Coding and Focused Coding phases, in order to make this final phase as painless as possible.

4.3 Dramatis Knowledge Engineering

This section details the implementation of the qualitative knowledge engineering methodology with respect to Dramatis. I begin by describing the survey that I used to generate my corpus. Following this, I describe the codes and guidelines that were created during Focused Coding, and the inter-coder agreement results for the third stage of coding. Finally, I describe the procedures that I used to convert the coded corpus into planning operators and scripts.

4.3.1 Knowledge Acquisition Survey

In order to generate the required natural language corpus required by Dramatis, it was first necessary to identify the stories that would be used during experimentation. After considering the taxonomy of suspense discussed previously (see Section 2.10), I selected three scenes, each of which was adapted from suspenseful scenes in popular films. The scenes selected are:

- From the film *Casino Royale* [15], the scene where James Bond is poisoned at the poker table and attempts to cure himself.
- From Alfred Hitchcock's film *Rear Window* [37], the scene where Lisa breaks into Thorwald's apartment to find evidence that Thorwald murdered his wife.
- From the film *Harry Potter and the Half-Blood Prince* [76], the scene where Draco attempts to assassinate Professor Dumbledore, while Harry watches from below.

Based on these movie scenes, I generated three survey prompts. The survey prompts described the beginning and end of one of the scenes. Respondents were instructed to list the steps that occurred in the story between these two points. I created two prompts from the *Casino Royale* example. The first (*Spy 1*) asked participants to describe how a spy could go from being in a bar to being poisoned.

Table 1: Knowledge Acquisition Study prompts

Spy 1	Start: A spy is at a bar or restaurant. Finish: The spy drinks from a drink poisoned by the villain.
Spy 2	Start: A spy is at a bar or restaurant. The spy just drank from a drink that was poisoned by the villain. Finish: The spy is no longer poisoned.
Rear Window	Start: A man (Tom) and a woman (Erin) suspect their neighbor of committing murder. Tom cannot leave the apartment, but Erin has just left the apartment to sneak into their neighbor's apartment to find proof. Tom and Erin have an agreed upon signal for if the neighbor is on his way home. Finish: The neighbor catches Erin in his apartment.

The second (*Spy 2*) asked participants to describe how the spy would subsequently be cured of this poison. For the *Rear Window* example, the prompt described a scene where two people suspected their neighbor of murder. One of these people was on their way to the neighbor's apartment to search for evidence. Participants were instructed to describe the events from entering the neighbor's apartment to being caught intruding by the neighbor. Because the *Harry Potter* example will demonstrate that Dramatis can use knowledge of character plans rather than scripts, no prompt was generated for that story. Additionally, there was concern about the possible breadth of answers in a domain that allowed for magical acts at almost any time. Each of the three prompts were written to avoid explicitly referring to its source material. In the *James Bond* prompts, the text referred to a generic spy. In the case of *Rear Window*, character names were changed. Table 1 shows the specific prompts given to participants.

Each prompt was placed in a Google web survey, with 20 numbered blank text fields. Participants were instructed to describe the events between the specified start and finish points by placing them in order in the text fields. The instructions also made clear that entries should focus on events or actions rather than setting. The

Table 2: Statistics of Knowledge Acquisition Study responses

Prompt	No. Responses	Total No. of Entries	Median Entries	Mean Entries (SD)
Spy 1	18	131	7	7.28 (3.78)
Spy 2	24	168	5	7.00 (4.79)
Rear Window	18	198	9.5	11.00 (5.11)

participants were instructed that they did not need to use all 20 fields.

Participants were directed to a webpage where all three surveys had been embedded in a random order. Participants were recruited using Georgia Tech mailing lists, contacts with other academic labs, and posts on Facebook and Twitter. The consent page informed subjects that they needed to be at least 18 years old to participate. Naturally, as a web survey, no actual age verification was possible.

4.3.1.1 Survey Results

While participants were asked to complete all three surveys, it is apparent that not all did. The *Spy 1* and *Rear Window* surveys received 18 responses, while *Spy 2* received 24 responses. Table 2 shows the total and average number of entries for each response in each survey.

4.3.2 Survey Corpus Coding

I coded the survey responses according to the Qualitative Knowledge Engineering methodology described above in Section 4.2.2. In the Focused Coding phase (Stage 2), the non-action codes generated in the first stage were reduced to a taxonomy of eleven codes that represented the space of possible rejection reasons. These eleven codes formed the basis of the coding guidelines that were used in the third phase. I separated codes for actions into two sub-categories: a code for entries representing single actions, and a code for entries representing multiple actions. Table 3 shows the categories created in this phase, as well as the guidelines for applying these codes in the third phase.]]

Table 3: Coding Guide for Knowledge Acquisition Responses

Code Type	**Shorthand**	**Description**
Single action	[Specify action]	Applies when the entry describes a single action/operator. Provide the operator in the response.
Multiple actions	[Specify actions]	Applies when the entry describes multiple actions/operators.
Prompt Failure	PROMPT	Applies when the end state of the response does not match the state instructed by the prompt.
Attention	ATTN	Applies when an entry deals with what a character is paying attention to or noticing.
Dialogue Action	DLG	Applies when an entry deals with what a character said. Does not apply when the entry just says two characters talked.
State	STATE	Applies when an entry provides state information but no action.
Thoughts	THGT	Applies when an entry deals with what a character is thinking or thinking about.
Inaction	INACT	Applies when an entry describes a character explicitly not taking an action.
Presentation	PRES	Applies when an entry describes audience point-of-view or sjuzet details.
Incomplete Actions	INC	Applies when a character begins performing an action or task but does not complete it.
Continuation	CONT	Applies when an entry is a continuation of the previous entry, or of the action described in the previous entry.
Continuing Failure	CF	Applies when an entry represents multiple attempts to do something with repeated failure and/or no expectation of immediate success and/or waiting for something to happen.
Vague	VAGUE	Applies when an entry says something happens, but not how; or when an entry provides multiple options for what might have happened.

For the third phase of coding, I randomly selected five responses from each prompt that would be coded independently by myself and a partner. These responses constituted 23% of the corpus. During this phase, the two coders agreed on 76.3% of codes (Cohen's $\kappa = 0.64$). Additionally, every time both coders marked an entry as an action, we independently reached semantic agreement on what action the entry represented. When we reduce codes to a simple Accept/Reject question (where Accept is a single action or multiple actions, and Reject is any of the eleven non-action codes), we agreed on 83.9% of codes ($\kappa = 0.67$). The difference in agreement rate and inter-rater reliability (Cohen's κ) comes from the latter taking into account the probability that coders agreed by random chance. The value for inter-rater reliability does not change drastically between the two cases because the likelihood of chance agreement is higher with only two options. Prior to this phase, we had agreed that "good" inter-coder agreement would be sufficient, so we only performed one iteration of this coding stage.

In the final phase of coding, I coded the remainder of the corpus using the same set of guidelines used in the previous phase. Additionally, I resolved coding disagreements from the third phase. When both coders indicated Reject, there was no need to resolve the conflict, as the ultimate result was the same. Re-coding was only necessary when one of the coders had indicated an action, while the other coded the entry as a rejected entry. Re-coding would also have been necessary if both coders agreed that an entry represented an action, but disagreed substantially on what this action was, but that never occurred. At the end of this process, every individual entry from each survey response had been tagged as one of the following:

- A single action, and what action is indicated.
- Multiple actions, and what actions are indicated.
- A candidate for rejection, along with the specific rejection code from Table 3.

4.3.3 Generating Planning Operators

Prior to converting the coded corpus to Dramatis knowledge representations, any survey response with an entry coded as "`Prompt Failure`" was removed from the corpus. This code indicated that the respondent had ignored the prompt in some way, typically failing to meet the specified conditions at either the beginning or end of the story. Thus, the knowledge provided in such responses was not applicable to Dramatis. Other rejection codes only affected the single entry, rather than the entire survey response.

After the completion of coding, each identified action was converted into a STRIPS operator [25]. Similar actions, or actions that were special cases of other actions (e.g., "sneak" is a special case of "go"), were combined into single operators. While the survey responses and the coding process provided the operation, or the verb, STRIPS operators require parameters, preconditions, and effects. None of these details could easily be parsed from the corpus. While some parameters can be determined based on the grammar of the corpus entry, such as the subject or objects of the sentence, additional parameters are sometimes needed to describe details that are implied by the natural language (e.g., The operator `give` requires a place parameter to ensure that both characters involved are co-located). Operator preconditions and effects were added based on how the actions were used in the corpus, and how they interacted with other operators during testing. Additionally, some operators were added that were not part of the crowdsourced materials, in order to make planning tasks viable. The operator sets created from each prompt were tested independently from the other prompts. Testing was considered complete when the Heuristic Search Planner (see Section 3.4.5.3) was capable of generating multiple plans that were part of the natural language corpus.

Appendix A shows the operator sets that were created from each prompt.

4.3.4 Generating Scripts

After operators were finalized, all survey responses corresponding to a particular prompt were incorporated into the relevant script. One script was created for each prompt. Each response to a prompt represented a portion of the script, a valid path through the script graph. Actions were added to the script with their corresponding operators. Rejected entries were skipped, unless doing so would break the coherence of the story. In these cases, an existing operator was typically relevant to the entry, unaffected by the rejection reason. Additionally, in some cases, events were included in the script trace that had been left implicit in the survey response (e.g., the operator `make-drink` was specified between `order` and `deliver-drink` by some respondents, but not all).

In addition to sequences of actions, Dramatis scripts also contain temporal links and causal links. Temporal links are directed edges that indicate that one action follows another temporally. A causal link exists when the effect of one script node operator achieves a precondition of a node that is a temporal descendant of the original node. Like temporal links, causal links are directed edges in the script graph. I create a temporal link between two events in the graph when one or more participants listed these events sequentially. In the cases where implicit events were added to the script trace (such as the example above), temporal links were created that went through the implicit event, rather than directly between the neighboring steps in the corpus. After temporal links were added, I added causal links between pairs of nodes when (a) the nodes could be on the same traversal of the script, and (b) the effects of one node satisfied the preconditions of the second node, where no node in the interim changed the state according to the proposition in question. As a result, a single precondition could be referenced by several causal links, each representing different traversals of the script graph.

Script definitions are broken into four portions. The first section lists the nodes

of the script graph, which contains only an identification number and an operator name. Only unbound operators are listed in the script definition. Information about operator parameters and how they relate to those of other script nodes is included later in the script. The second section is an adjacency list for the temporal links for the script graph. The third section lists the co-designation constraints for operator parameters. For example, if the same character performed two different actions in a survey response, leading to two separate nodes in the script, I add a co-designation constraint for the actors of these two script nodes. This tells Dramatis that if it seems one of these actions, it should expect that the same character performs the second action. Not all operator parameters are co-designated with other parameters. The final section of the script definition describes the locations of causal links in the script.

Appendix A contains the scripts and visualizations of each script.

4.3.5 Discussion

In future work, it would be beneficial to expand the third phase of the coding process. While an inter-rater reliability score of 0.64 is considered "good" agreement, "excellent" inter-rater reliability is typically indicated by $\kappa \geq 0.8$ [43, 20, 26, 53]. Additional iterations would provide greater confidence for the results of the single coder in the final coding phase. However, it is notable that, as calculated, Cohen's κ does not account for the agreement achieved in specifying which actions were being described in each entry. Rather, it only tracked when both coders indicated `Single action` or `Multiple actions`. Accounting for the level of agreement in action descriptions might increase the inter-coder agreement coefficient. Ultimately, for the purposes of Dramatis, I was satisfied with the raw percentage of agreement, as well as the Cohen's κ coefficient indicating "good" agreement.

It is important to recognize that the codes developed in this implementation of the qualitative knowledge engineering methodology, while applicable across all domains

given in my prompts, are not necessarily applicable to all knowledge-engineering tasks. Rather, other researchers would have to go through the same processes that I used to properly identify the appropriate codes, allowing the codes to emerge from the dataset. It is easy to imagine other systems that would have different criteria for determining which responses were acceptable and which were not. For example, other researchers may want to include dialogue actions (where I did not) or would want to exclude responses that I deemed acceptable. Ultimately, these are the types of decisions that must be made prior to coding or in the initial phases of the coding process, depending entirely on the goals of the researchers.

4.4 Conclusions

The qualitative knowledge engineering methodology presented in this chapter is applicable to any knowledge engineering task that attempts to convert natural language into a knowledge representation. This process allowed for the collection of a corpus of natural language descriptions of events that could be turned into a set of planning operators and scripts. By ensuring that my system has knowledge collected from typical human readers, I am better able to compare its output to the suspense ratings given by humans. Without this knowledge, this type of direct comparison between humans and Dramatis would be impossible. By collecting this knowledge from a variety of people, I was able to remove some personal biases that otherwise could have influenced the knowledge engineering in my selected domains. While I was heavily involved in the coding process, the use of a second coder helped ensure that appropriate decisions were made in that process. Additionally, the knowledge acquisition and coding processes described here are easily transferable to other domain engineering tasks in artificial intelligence. Further use of this methodology, in the contexts of other systems and other knowledge representations, could indicate additional benefits of this approach as compared to other knowledge acquisition techniques.

CHAPTER V

EVALUATION

In this chapter, I report on three evaluations of Dramatis. In Evaluation 1, I present Dramatis and human readers with the same six stories. Dramatis produces suspense curves, while human participants rate the suspensefulness of each story. I compare the model results to the ratings given by human readers. In Evaluation 2, I ablate the functionality of the MEI-P Situation Model, such that it is no longer used in calculating the cost of planning operators, in order to demonstrate the necessity of the memory model to Dramatis. Finally, in Evaluation 3, I ablate the algorithm for determining goal states for escape planning, with the intention of demonstrating that the technique used for establishing the escape planning problem is a necessary step in this suspense model.

5.1 Materials

For the evaluations of the computational suspense model, I needed pairs of stories that have different levels of suspense. Each pair contains an original version and an alternate version of the story, where details were changed in a way that I believed would lower the overall perception of suspense. Dramatis was given computational versions of these stories in the form of Time-Slices, while human participants received versions in natural language.

The three original stories were adapted from scenes from the films *Casino Royale* [15], *Rear Window* [37], and *Harry Potter and the Half-Blood Prince* [76]. Each alternate version was crafted with the intent of reducing the suspense level according to the model. *Casino Royale* and *Rear Window* were selected as examples of *genre-knowledge suspense* from my categorization of suspense (see Section 2.10). *Harry*

Potter was selected for these evaluations as an example of *opposition suspense*. Appendix B contains the natural language and Time-Slice versions of all six stories. However, for summarization purposes, I will now briefly explain the scenes in question and the differences between the original and alternate versions of each story pair.

5.1.1 Casino Royale

The film *Casino Royale* [15] was adapted from Ian Fleming's first novel featuring James Bond, by the same name. In their description of suspense, Gerrig and Bernardo use a scene from that book where a gun is placed at James Bond's back while he is at a baccarat table, and he must devise an escape [30]. In the film, the scene is altered so that Bond receives a poisoned drink while he is at a poker table in the midst of a high-stakes tournament. Once he realizes that he has been poisoned, Bond attempts to cure himself, first by vomiting, and then by using an antidote and a defibrillator in his car. Unfortunately, Bond passes out before he can defibrillate himself, but he is rescued by his ally, Vesper Lynd, just before he dies.

The alternate version makes the following changes to the story:

- The reader is informed of the existence of the medicine and defibrillator earlier in the story.
- The sequence in which Bond attempts to vomit to cure himself is removed.

By introducing the medicine and defibrillator sooner, we provide the reader with additional escape routes for James Bond that did not exist in the original version. By removing the vomit sequence, which fails to cure Bond in the original version, we prevent the removal of a potential escape plan, thus maintaining the quantity of escape plans.

5.1.2 Rear Window

Rear Window [37] was written and directed by Alfred Hitchcock, widely considered to be the master of suspense in film. In this story, the protagonist, a photographer named Jeff, has broken his leg and is unable to leave his wheelchair in his New York City apartment. Jeff uses his camera's telephoto lens to spy on his neighbors, eventually coming to the conclusion that one of them (Thorwald) has murdered his wife. In the scene used in this study, Jeff's friend Lisa sneaks in to Thorwald's apartment to find proof of the murder. However, Thorwald returns home before she is able to escape. Thorwald begins to beat Lisa, as she tries to explain herself. Jeff watches from his apartment with his friend Stella, neither of whom are able to do a thing until the police arrive.

The alternate version makes the following changes to the story:

- Jeff calls the police about Lisa and Thorwald sooner.
- Stella is removed entirely.

Introducing the police earlier in the story provides the means for an escape plan earlier than in the original version. By removing Stella, we remove a character that may be distracting, thereby giving increased weight to other characters in the memory model. I believe that other characters will now be more strongly activated, making escape plans using those characters cheaper.

5.1.3 Harry Potter and the Half-Blood Prince

The film *Harry Potter and the Half-Blood Prince* [76] is based on the J.K. Rowling novel of the same name. Throughout the novel, one student, Draco, has been making attempts to kill school headmaster, Dumbledore. One of the teachers, Professor Snape, has been sworn to assist Draco should he fail. At the climax of the story, Harry Potter and Dumbledore return to the school. Dumbledore orders Harry to

hide, while Draco confronts the headmaster. Draco quickly disarms Dumbledore of his wand and tries to work up the nerve to kill him. Meanwhile, Professor Snape arrives, whom Harry believes is here to help. Snape orders Draco to step aside, and ultimately kills Dumbledore himself.

Unlike the other two stories, which make use of scripts in Dramatis to represent reader knowledge, I present Dramatis with Draco's plan to kill Dumbledore in the place of scripts.

The alternate version makes the following changes to the story:

- Draco does not disarm Dumbledore of his wand.
- Rather than pleading with Snape, Dumbledore aims his wand at Snape just before being killed.

Suspense is lowered in the alternate version because Dumbledore retains his wand. Dumbledore remains capable of saving himself, rather than relying on other characters, like Harry or Snape. Further, by stating that Dumbledore aims his wand, we remind readers of its existence and possible use.

5.1.4 Dramatis Input

In addition to the story in question, Dramatis is given scripts and operators as part of its input. For *Casino Royale* and *Rear Window*, the scripts and operators given to Dramatis (or an ablated form of Dramatis in Evaluations 2 and 3) were created using the knowledge acquisition and engineering methodology described in Chapter 4. In the case of *Harry Potter and the Half-Blood Prince*, I defined the operators myself, while using a character plan instead of scripts to represent audience knowledge. The operators, scripts, and character plan used for each story can be found in Appendix A.

5.2 Evaluation 1: Human-Dramatis Comparison

The purpose of Evaluation 1 is to compare the suspense ratings produced by Dramatis to human self-reported ratings of suspense for the same stories. Dramatis is given the stories described above in Time-Slice format, while human participants read natural language versions of the stories. More specifically, both Dramatis and human participants will be given stories in pairs, and each will be asked to identify the more suspenseful story.

Hypothesis 1: Dramatis orders stories according to suspense level the same way humans order stories according to suspense level.

If this hypothesis holds, we can conclude that Dramatis perceives the suspensefulness of entire stories the same way that humans do.

5.2.1 Method

I recruited human participants to read natural language versions of the six stories described above in Section 5.1: Original and alternate versions of *Casino Royale*, *Rear Window*, and *Harry Potter and the Half-Blood Prince*. All participants read the story pairs in the same order (*Casino Royale*, *Rear Window*, *Harry Potter*), but I controlled for order within pairs (original vs. alternate). After reading each individual story, participants answered "How suspenseful was this story?" on a 7-point Likert scale. Additionally, after each pair of stories, participants selected which of the pair was the more suspenseful story, thus producing an ordering for the two stories according to suspensefulness. Participants were not able to see their previous Likert responses while answering questions about later stories.

Two additional questions were added while the study was being run to check reader comprehension. After reading all stories, participants were asked "What differences, if any, existed between Story C and Story D?", referring to the *Rear Window* stories. This was followed immediately by an identical question for the *Harry Potter* stories.

Table 4: Results of Evaluation 1 preference questions

Story	Original Version	Alternate Version	Wilcoxon Signed-Rank	Statistical Significance
Casino Royale	31	1	16.5 ($z = 4.62$)	$p < .001$
Rear Window	18	14	231 ($z = 0.61$)	n. s.
Harry Potter	23	9	148.5 ($z = 2.16$)	$p < .05$

These questions were added because anecdotal evidence suggested that early subjects were not able to accurately describe the differences between the *Rear Window* or *Harry Potter* stories. Appendix B contains the materials used in this study.

Dramatis is given the same six stories in time-slice format, rather than natural language. For Dramatis, I measure the overall suspense level of a given story by calculating the area under the suspense curve (AUC). When comparing two stories of differing lengths, I also report the average area under the curve—that is, AUC divided by the number of time-slices.

5.2.2 Results

Thirty-two people participated in this study, 16 of whom answered the additional questions about story comprehension. Table 4 shows the results of the simple preference questions for each pair of stories. This table shows the number of people who selected each version of the story, as well as the Wilcoxon Signed-Rank test result, the z-score, and the one-tailed statistical significance level. In general, participants found significant differences in the suspensefulness of *Casino Royale* and *Harry Potter*. While more participants rated the original version of *Rear Window* to be the more suspenseful version, the difference was not statistically significant.

Table 5 shows the results of the Likert-scale questions for each individual story. The columns for the two versions show the interpolated median Likert rating. As above, the table shows the results of the Wilcoxon Signed-Rank test, and one-tailed statistical significance. The column labeled n_r indicates the number of participants

Table 5: Results of Evaluation 1 Likert-scale questions

Story	Interpolated Median Rating		n_r	Wilcoxon Signed-Rank (z-score)	Statistical Significance
	Original	Alternate			
Casino Royale	5.59	3.23	30	5 ($z = 4.67$)	$p < .001$
Rear Window	4.90	4.73	28	121 ($z = 1.86$)	$p < .05$
Harry Potter	4.83	4.14	19	28 ($z = 2.69$)	$p < .01$

Table 6: Results of Evaluation 1 reading comprehension questions

Story	Detected Both Chgs.	Detected One Chg.	Detected Neither	Wrong Change	Unsure
Rear Window	3	10	3	2	1
Harry Potter	0	8	8	9	3

who gave differing ratings for the two versions. For all three stories, human readers ratings were significantly higher for the original version than for the alternate version.

Table 6 describes the results of the reading comprehension questions. The first three columns count how may participants detected both, one, or neither of the changes in the stories. The "Wrong Change" column indicates how many people gave an answer that included at least one difference that was not actually present between the two stories. The "Unsure" column indicates the number of people who could not recall the difference or were uncertain that there was a difference between the stories.

All statistics reported in this section referring to human participants apply the Wilcoxon signed-rank test (W). When not indicated, $n = 32$. When n_r is given for Likert-scale questions, it indicates that the remaining participants gave equal Likert ratings for the both versions of the story. In all cases, a conversion from W to z-scores are provided, and one-tailed significance values are also given, when appropriate.

5.2.2.1 Casino Royale

For *Casino Royale*, 31 of the 32 participants (96.8%) selected the original version as the more suspenseful version of the story ($W = 16.5$, $z = 4.62$, $p < .001$). Additionally, Likert-scale ratings favored the original version of the story, with a median rating for the original version of 6 (on a 7-point scale), and a median rating of 3 for the alternate version ($n_r = 30$, $W = 5$, $z = 4.67$, $p < .001$). Thus, human readers showed a strong preference for the original version of the story according to both measures.

Figure 15 shows the suspense ratings produced by Dramatis for the original version (shown in blue) and the alternate version (shown in orange) of *Casino Royale*. Because the original version has more Time-Slices, the graph in Figure 15 has been arranged so that corresponding events between the two stories are aligned. When a Time-Slice in the original story version has no corresponding Time-Slice in the alternate version, the line representing the alternate version is interpolated based on the suspense ratings for the two neighboring Time-Slices.

In general, Dramatis finds the original version more suspenseful than the alternate version of *Casino Royale*. Most notably, in the original version, Dramatis provides a maximum suspense rating for Time-Slices 9-11. In Figure 15, maximum suspense is indicated by a value of 5000 units. At this point in the story, Bond has failed in his attempt to vomit the poison out of his body, and no other solution has been indicated in the story. Dramatis does find the alternate version to be more suspenseful at certain points. For example, at Time-Slice 7, when Vesper sees Bond leave the poker table, Dramatis finds the alternate version more suspenseful. While the graph shows that the alternate version is also more suspenseful in the neighboring Time-Slices, these values were interpolated for missing events. Dramatis also rates the alternate version marginally more suspenseful in Time-Slices 12-14. However, when Dramatis finds the original version to be more suspenseful, the difference between suspense ratings is

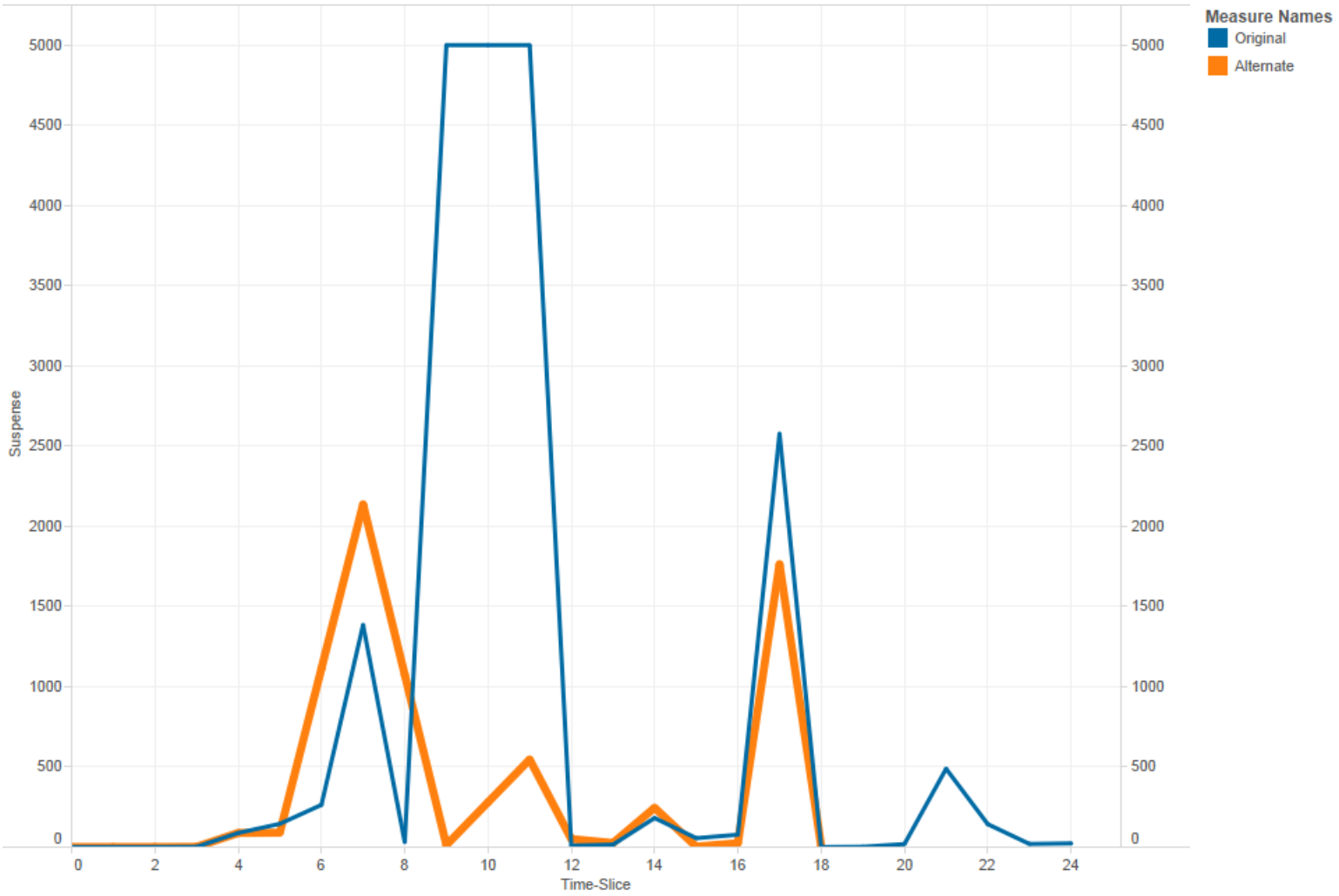

Figure 15: Dramatis suspense curves for *Casino Royale* stories

much greater than when the reverse is true.

Using area under the curve (AUC), it is clear that Dramatis finds the original version more suspenseful. Over its 25 Time-Slices, the original version has an AUC of 20526 square units, for an average of 855.3 units per time-slice. By comparison, the AUC for the alternate version is 7468 square units, for an average of 414.9 units per time-slice. Reducing the original version to its first 19 Time-Slices (the same length as the alternate version), the average area is 1102.2 units, more than 2.5 times the area under the curve for the alternate version. Thus, the original version of the *Casino Royale* story is rated more suspenseful than its alternate by both Dramatis and human readers.

5.2.2.2 Rear Window

In the *Rear Window* case, 18 of the 32 participants (56.3%) selected the original version as being more suspenseful ($W = 231$, $z = 0.61$, not significant). However, Likert-scale ratings showed more of a preference for the original version. The interpolated median rating for the original version was 4.90, and the interpolated median for the alternate version was 4.73 ($n_r = 28$, $W = 121$, $z = 1.86$, $p < .05$). Additionally, an ordering effect was present for this story pair. Of the 32 participants, 22 (68.8%) indicated the version they read first was the more suspenseful version ($W = 165$, $z = 1.85$, $p < .05$), while 20 (62.5%) participants gave a higher Likert-scale rating to the first version they read ($n_r = 28$, $W = 142$, $z = 1.38$, not significant). Overall, human readers favored the original version of *Rear Window*, but the difference in ratings was significant only for the Likert-test, which may have been influenced by an ordering effect.

Additionally, of the 16 participants who were given reading comprehension questions, 13 of them (81.3%) successfully identified at least one of the differences between the two *Rear Window* versions. Eight respondents recognized that Stella was only

present in one version. However, each of the participants who identified that difference read the version without Stella (the alternate version) first.

Figure 16 shows the suspense curves produced by Dramatis for the original (blue line) and alternate (orange) versions of *Rear Window*. Three points in the suspense curve for the alternate version are interpolated to cover events that were removed from the original version.

While neither story is considered particularly suspenseful for the first half (note that the stories are identical for Time-Slices 0-7), the original version clearly becomes suspenseful at the moment that Lisa is caught by Thorwald (Time-Slice 16). Meanwhile, because of the earlier introduction of the police (at Time-Slice 8, rather than Time-Slice 14), and the fact that Stella is no longer present in the memory model, the alternate version of the story never becomes suspenseful, according to Dramatis.

Looking at AUC for each suspense curve, Dramatis finds the original version of *Rear Window* to be more suspenseful. The total AUC for the original is 4176 square units, as compared to just 1170 square units for the alternate version. The vast majority of this difference occurs between Time-Slice 16 (Lisa's capture) and the end of the story. Thus, Dramatis finds the original version of *Rear Window* to be more suspenseful than the alternate version.

In summation, Dramatis rates the original version more suspenseful than the alternate version, while human readers are conflicted. Likert ratings for the original version of *Rear Window* are significantly greater than those of the alternate version, but when forced to make a single choice, human raters do not show a statistically significant preference for either version. Additionally, human ratings may have been influenced by which version they read first.

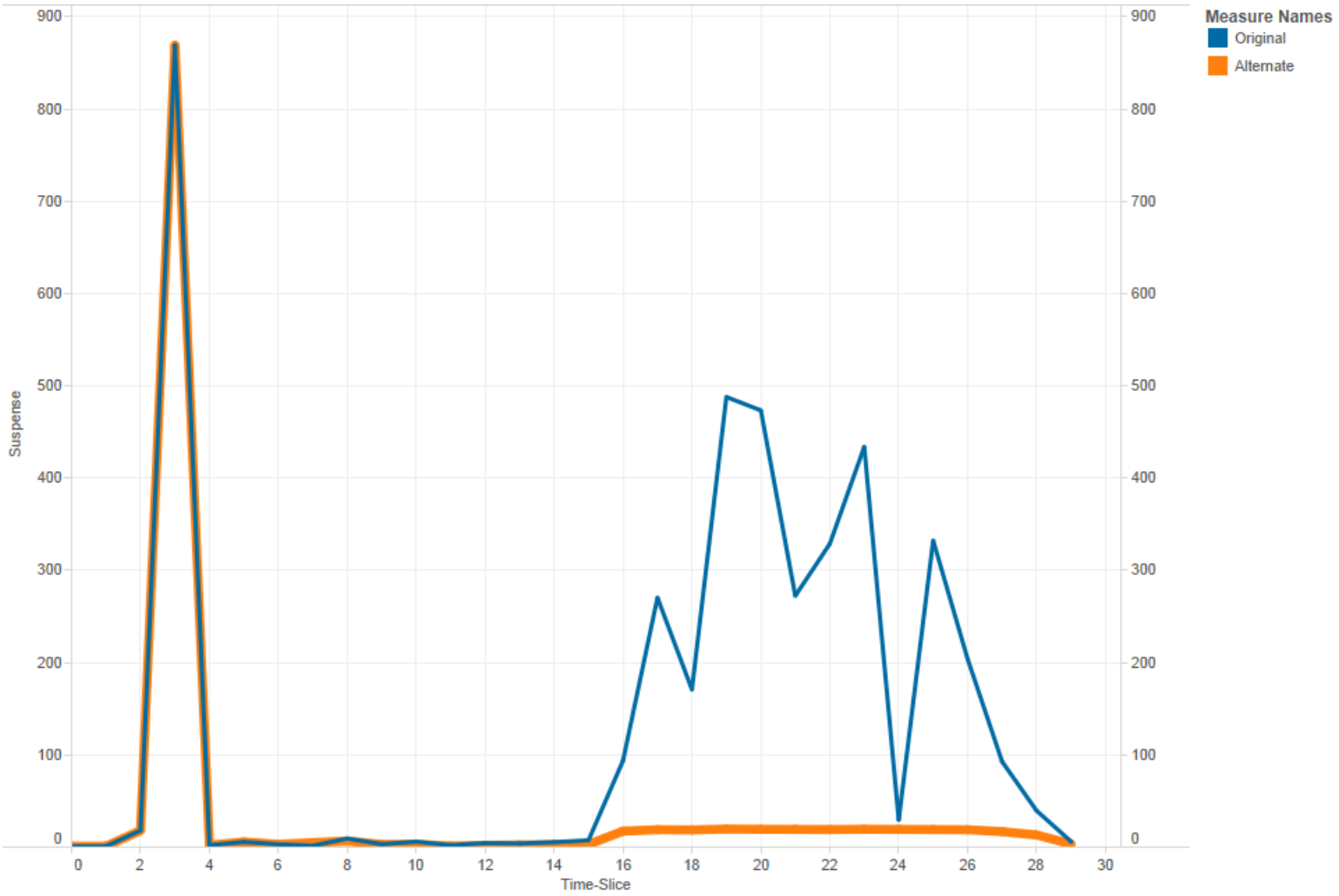

Figure 16: Dramatis suspense curves for *Rear Window* stories

5.2.2.3 Harry Potter and the Half-Blood Prince

Finally, for *Harry Potter*, 23 of the 32 participants (71.9%) selected the original version as being the more suspenseful version of the story ($W = 148.5$, $z = 2.16$, $p < .05$). Of the 32 participants, 19 gave different ratings for the two versions in the Likert-scale questions. The median Likert-scale rating for the two versions were 5 and 4 for the original and alternate versions, respectively ($n_r = 19$, $W = 28$, $z = 2.69$, $p < .01$). Unlike *Rear Window*, no ordering effect was present, as 17 of 32 participants (53.1%) selected the version they read first. In summation, participants self-reported suspense levels were significantly higher for the original version of *Harry Potter*.

Sixteen participants were given additional reading comprehension questions to identify the differences between the two *Harry Potter* versions. Only 8 of the 16 participants successfully identified the differences between the two stories. However, 9 of the 16 participants responded with differences between the versions that were not actually present.

Figure 17 shows the suspense curves for the *Harry Potter* stories. The suspense curve for the original version is shown in blue, while the curve for the alternate version is shown in orange. The curves separate when Dumbledore is disarmed by Draco in the original story (Time-Slice 10). For the original story, Dramatis is less able to generate escape plans wherein other characters can disarm Draco. In the alternate version, Dramatis continues to find escape plans where Dumbledore saves himself by disarming his would-be assassin. Unlike the previous pairs of stories, no values in the alternate version curve are interpolated from neighboring values.

As with *Casino Royale*, there is a brief sequence where Dramatis finds the alternate version of *Harry Potter* to be more suspenseful than the original version (Time-Slices 19-25). However, the differences between the versions are relatively small compared to the sections of the story where Dramatis finds the original to be more suspenseful.

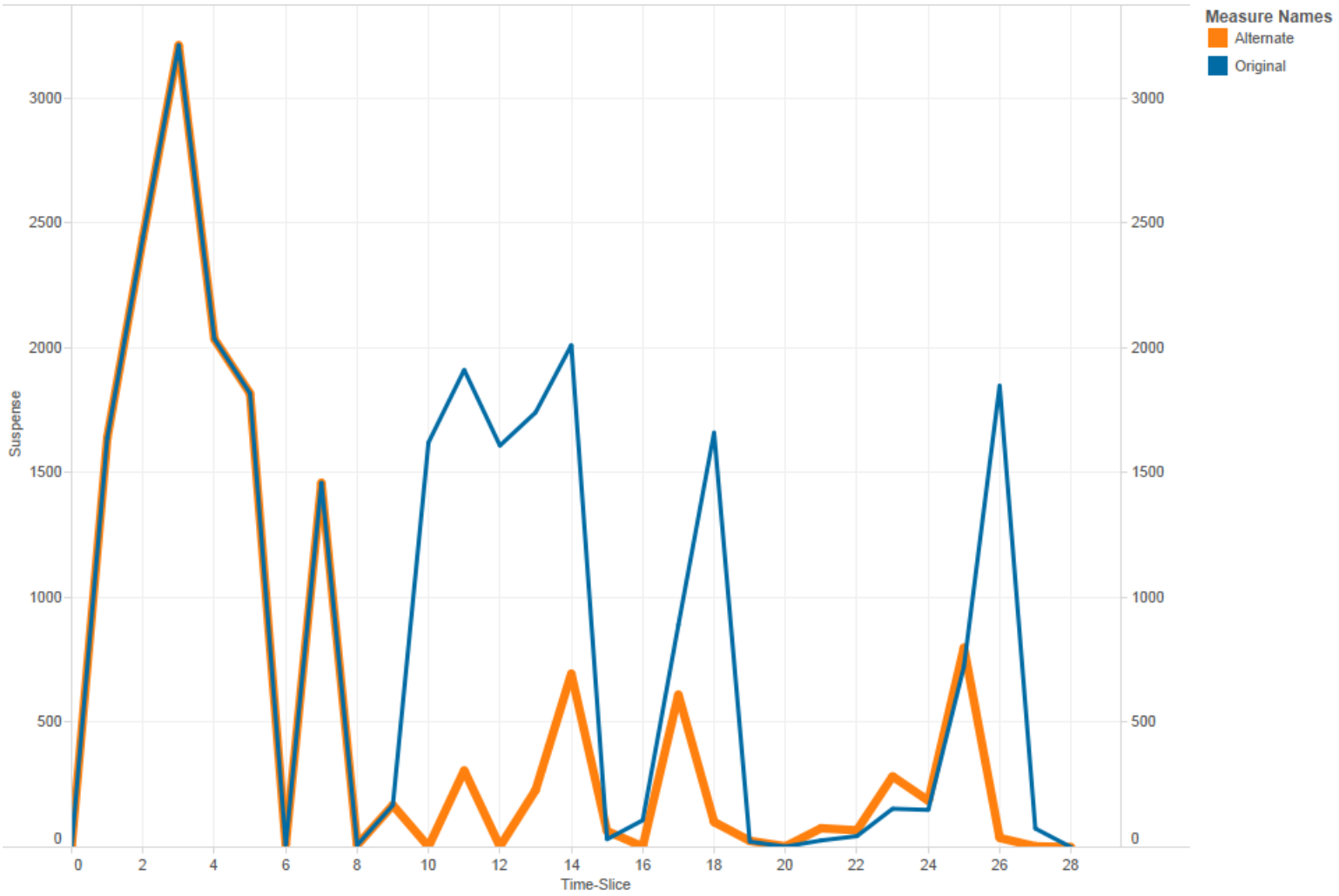

Figure 17: Dramatis suspense curves for *Harry Potter* stories

Notably, this occurs in Time-Slices 10-18 (from Dumbledore being disarmed to a moment when Harry aims his own wand), and from Time-Slice 26 (wherein Dumbledore either pleads with Snape, or aims his wand at him, depending on the version) to the end of the story.

When comparing area under the suspense curves, Dramatis finds the original version of *Harry Potter* to be more suspenseful. The total AUC for the original is 27428 square units, while the alternate version has an AUC of 16293 square units. This difference shows that Dramatis rates the original version as the more suspenseful of the pair.

Overall, Dramatis and human participants agree that the original version of *Harry Potter* is more suspenseful than the alternate version. Human readers showed a statistically-significant preference for the original version when asked choose which of the pair was more suspenseful. When asked to provide Likert ratings for each version, human participants continued to show a statistically significant preference for the original version, although these results are mitigated somewhat by the number of participants who provided the same Likert rating for both stories.

5.2.3 Discussion

I hypothesized that Dramatis would provide the same rankings for the suspensefulness of story pairs as human readers. In the case of *Casino Royale*, we clearly see agreement between Dramatis and human participants. With *Rear Window*, Dramatis agreed with the majority of human readers, although the result for human readers was significant only when considering Likert-scale ratings. Finally, Dramatis agrees with human readers that the original version of *Harry Potter* is the more suspenseful version.

The *Casino Royale* selection proved to be an easy decision for most participants. This pair had the most obvious changes, and the difference in suspense was apparent

to almost all participants. The one participant who selected the alternate version of *Casino Royale* also selected the alternate versions of both other pairs. These results may not seem surprising, because these story changes were perhaps the most obvious of the three story pairs. However, it is encouraging that Dramatis is correct about human feelings of suspense when humans find the answer obvious. Having the *Casino Royale* stories as the first pair also proved beneficial, as it allowed participants to self-calibrate their suspense ratings for the remainder of study.

While Dramatis considered the original version of *Rear Window* to be more suspenseful than the alternate version, the response from human participants was more nuanced. As noted above, participants did not overwhelmingly select the original version of *Rear Window* as the more suspenseful version. However, when their Likert-scale ratings indicated a difference, there was a statistically-significant preference for the original version. This significance comes from the fact that when participants preferred the original version, they were more likely to prefer it by a larger margin than those who found the alternate version more suspenseful.

The most glaring issue with *Rear Window* is the ordering effect. A statistically significant number of participants stated that the first version they read was the more suspenseful version. A related issue comes from the results of the reading comprehension questions. Of the 16 respondents to those questions, 8 reported recognizing that Stella was only present in one version. All eight of these participants read the version without Stella first (the alternate version). This would seem to indicate that readers who saw Stella initially did not believe she was an important character and, as a result, were not considering escape plans that involved Stella. It should be noted that Dramatis found the alternate (without-Stella) version less suspenseful, in part, because her absence allowed the existence of the police to be more prominent in the memory model. These results seem to indicate that the memory model provides too much weight to a character who human readers appear to consider insignificant to the

plot or the protagonist's well-being. Given the ordering effect, one possible future test for Dramatis would be to provide it with the *Rear Window* stories without clearing the memory model in between stories.

Of the two *Harry Potter* versions, Dramatis rates the original version to be more suspenseful. This decision was largely echoed by human readers. When forced to choose between the pair of *Harry Potter* stories, a statistically significant number of them chose the original version as the more suspenseful. However, 13 of the 32 respondents provided equal Likert-scale ratings for the two versions. Like *Rear Window*, when different Likert-scale ratings were given, participants stated a preference for the original version by a larger margin than those who believed the alternate version was more suspenseful. This difference in magnitude leads to the statistically significant Likert result.

A further item of interest is the number of people who incorrectly identified the differences between the two versions of *Harry Potter*, as it corresponds well with the people who gave equal Likert results. It is interesting that people still found the original version to be more suspenseful and were able to justify the choice in their responses to the reading comprehension question, even when the differences they described were in their own mind. Despite not being able to identify the differences, participants still believed the original version was more suspenseful, even when they could barely distinguish between the levels of suspense in the two stories. This may indicate that people consider likelihood of escape at a level they do not realize.

In summation, both Dramatis and human readers consider the original version of each pair of stories to be more suspenseful than the alternate version. When presented with the simple preference question, human readers showed significant preference for the original versions of both *Casino Royale* and *Harry Potter*. While readers also tended to find the original version of *Rear Window* to be more suspenseful, the difference in selection was not statistically significant and was possibly influenced by the

order in which the pair of stories was read. In the Likert questions, human readers rated the original versions of all three pairs to be significantly more suspenseful than the alternate versions. In general, these ratings are consistent with the output of the Dramatis system. Therefore, I can conclude that Hypothesis 1 has been supported. Human participants provide the same ordering according to suspensefulness as Dramatis, suggesting that Dramatis perceives the suspensefulness of whole stories the same way as human readers.

5.3 Evaluation 2: Ablated Memory Model

Evaluation 2 is an ablative test of the Dramatis model. In this study, the MEI-P Situation Model is removed from the planning process. I refer to this modified system as Dramatis-No-MEI. The MEI-P Situation Model represents the salience of story elements in the reader's mind. The activation of nodes in the situation model informs the cost of planning operators during the escape planning process, allowing Dramatis to identify the escape plan that would come to mind first for a human reader. Without MEI-P, Dramatis has no model of salience, and thus no conception of what story elements would be foregrounded in a reader's mind. Rather than having variable costs based on node activations within the memory model, all planning operators in Dramatis-No-MEI have costs of 1.0. Thus, with this ablation, I hope to demonstrate that the salience model is necessary for producing suspense ratings that are comparable to those of human readers.

> **Hypothesis 2:** Dramatis-No-MEI will produce an ordering of stories according to suspensefulness that is different from the ordering produced by Dramatis and human readers.

If this hypothesis holds, we can conclude that the MEI-P salience model is a necessary component for Dramatis to produce suspense ratings that agree with those provided by human readers.

5.3.1 Method

Other than the removal of the MEI-P Situation Model, the procedure for Evaluation 2 is identical to that of Evaluation 1 for Dramatis. Dramatis-No-MEI is given the same three pairs of stories, and the same scripts and operators as before. Suspense curves, by examining the area under the curve (AUC), will be compared to the suspense ratings produced by human readers in Evaluation 1.

5.3.2 Results

5.3.2.1 Casino Royale

Figure 18 shows the suspense ratings produced by Dramatis-No-MEI for the *Casino Royale* stories. The original version is shown in blue, with the alternate version in orange. As with Evaluation 1, the graph in Figure 18 has been arranged so that corresponding events between the two stories are aligned. When a Time-Slice in the original story version has no corresponding Time-Slice in the alternate version, the line representing the alternate version is interpolated based on the suspense ratings for the two neighboring Time-Slices.

For a substantial portion of the story, Dramatis-No-MEI provides equal suspense ratings for both versions of *Casino Royale*. It gives a maximum suspense rating (6 units) for the original version for Time-Slices 9-11. As mentioned above, this occurs when the model is unable to find a plan after James Bond fails in his first attempt to cure himself. Recall that this sequence is removed in the alternate version of the story. However, the only other separation between the two versions of the stories occurs at the end of the alternate version (Time-Slice 18).

When considering area under the suspense curve, Dramatis-No-MEI indicates that the original version is more suspenseful. Over the first 19 Time-Slices, the original version has a total AUC of 35.5 square units, compared to only 24 square units for the alternate version. A significant portion of this difference can be explained by

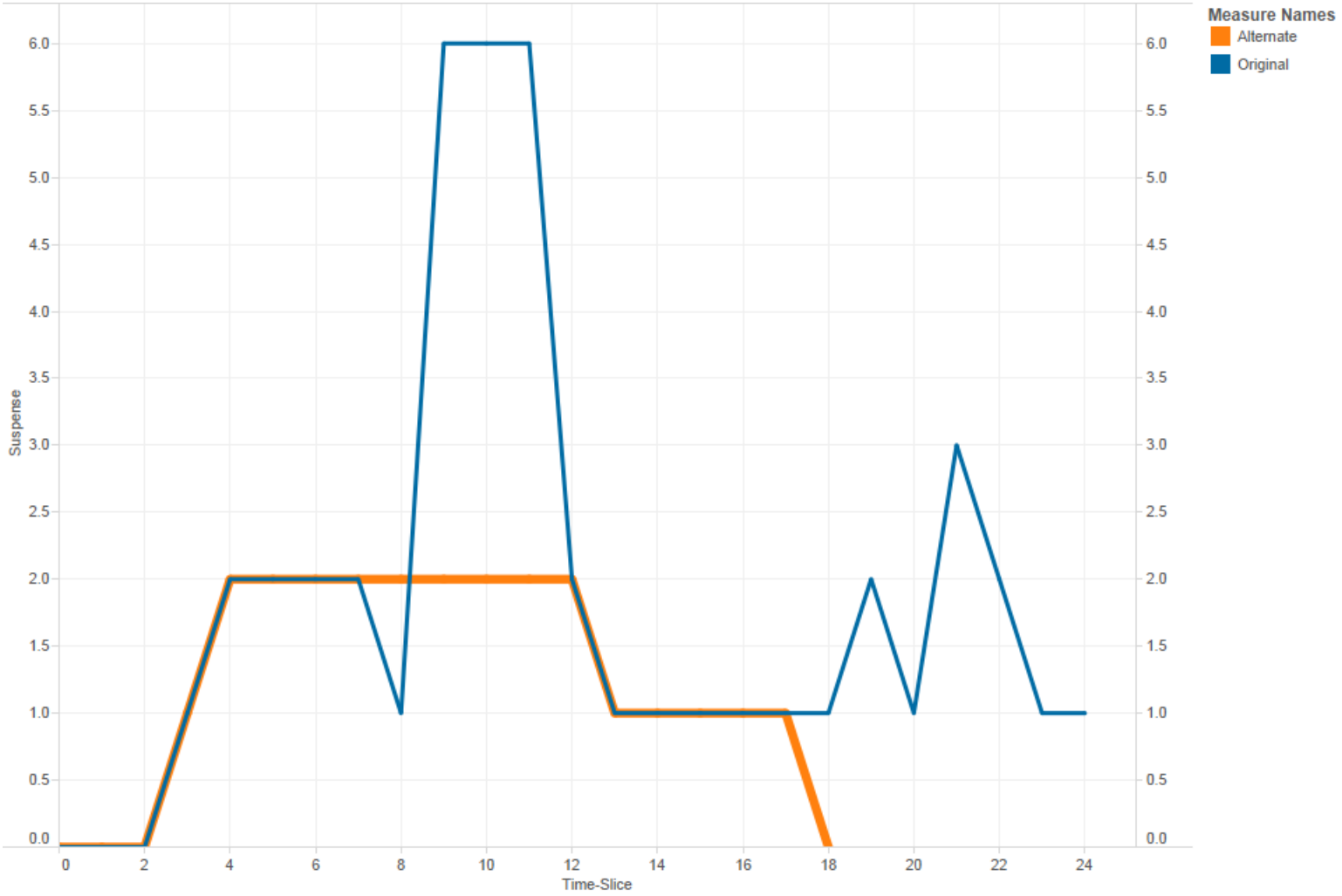

Figure 18: Suspense curves for *Casino Royale* stories from Evaluation 2

the section of the story where the model fails to generate an escape plan for Bond. For the remainder of the stories, the suspense ratings are identical. Thus, while the area is greater for the original version, the fact that this can be attributed to a single sequence indicates that the MEI-P situation model is a sufficient component for distinguishing between the two stories.

5.3.2.2 Rear Window

Figure 19 shows the suspense curves for Dramatis-No-MEI when given the *Rear Window* stories as input. As with results shown above, some suspense ratings in the graph for the alternate version are interpolated to cover events that were removed from the original version.

As can be seen in Figure 19, Dramatis-No-MEI can find no difference between the two stories. While the generated escape plans are different, the costs of those plans according to Dramatis-No-MEI are identical, regardless of the changes in the stories. Thus, without the MEI-P situation model, the model is unable to distinguish between the suspense levels of these two stories.

5.3.2.3 Harry Potter and the Half-Blood Prince

Figure 20 shows the suspense curves generated by Dramatis-No-MEI for *Harry Potter and the Half-Blood Prince*. The curve for the original version (shown in blue) indicates higher suspense than the alternate version (shown in orange) starting with the divergence between the stories at Time-Slice 10 (when Draco disarms Dumbledore). When comparing AUC, the original version has a total area of 64 square units, while the alternate version has a total area of 50 square units. Thus, removing the memory model did not change the relative suspense ratings for this pair of stories.

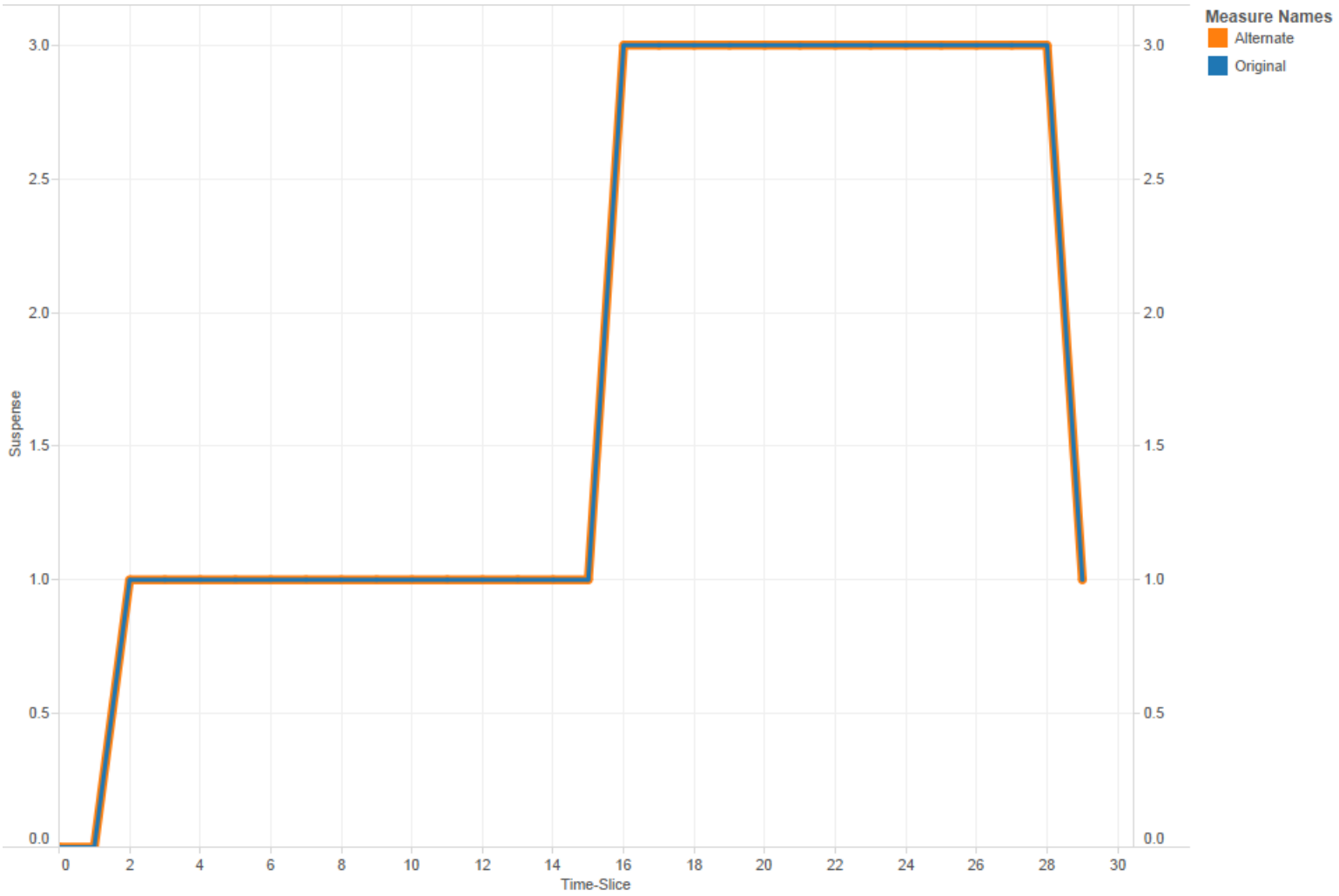

Figure 19: Suspense curves for *Rear Window* stories from Evaluation 2

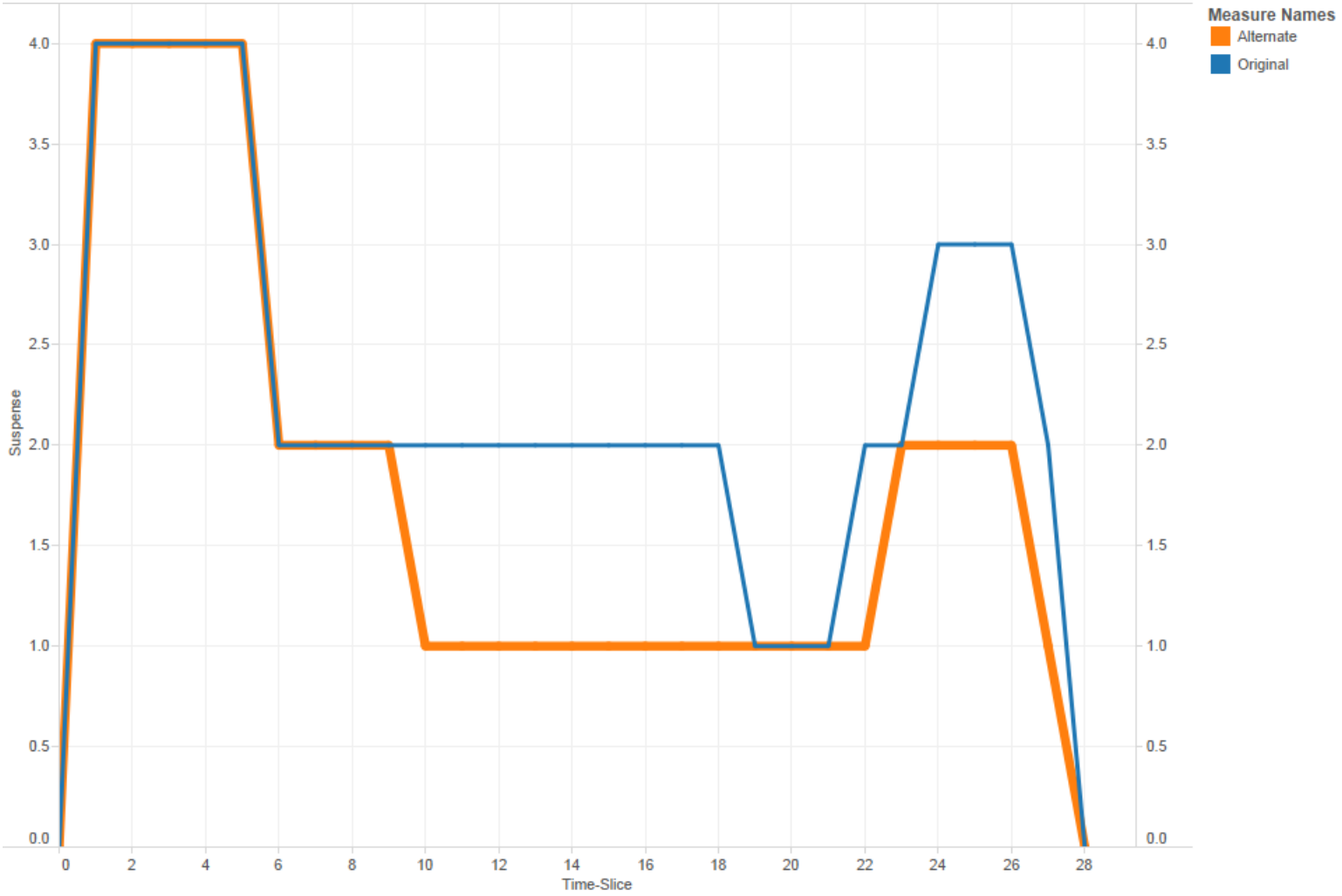

Figure 20: Suspense curves for *Harry Potter* stories from Evaluation 2

5.3.3 Discussion

In each pair of stories, I expected to see that Dramatis-No-MEI would either generate similar ratings for the two stories, or ratings that indicated that the alternate version was more suspenseful. Dramatis-No-MEI finds that the original version of *Casino Royale* is more suspenseful than its alternate version. While most of this can be attributed to a space of three Time-Slices, it is enough to create a disparity between the two stories. With the exception of that three slice section, however, we see similar ratings. This indicates that the MEI-P model is sufficient, if not necessary, for generating escape plans and suspense ratings. The results for *Rear Window* match the expectations for Dramatis-No-MEI. As described above, the suspense model cannot identify any difference between the two versions. Without the MEI-P situation model, Dramatis is unable to identify even the limited differences in suspensefulness that human readers identified in Evaluation 1. Unlike the other two stories, the results for *Harry Potter* do not match the predicted hypothesis for Evaluation 2. Without the memory model, Dramatis-No-MEI continues to find the original version of *Harry Potter* to be more suspenseful.

Hypothesis 2 has been partially supported. The ratings generated by Dramatis-No-MEI for *Rear Window* show a clear difference from those produced by Dramatis and human readers. Additionally, the *Casino Royale* and *Harry Potter* examples demonstrate that Dramatis-No-MEI cannot show the same gradations in the differences in suspense levels as the full Dramatis model. This suggests that Dramatis would be more effective at identifying less obvious differences in suspense only with the MEI-P Situation Model. Without it, any escape plans of equal length would indicate equal suspense levels, limiting the system's ability to distinguish suspense. Given these results, I conclude that the MEI-P Situation Model is a sufficient, but not necessary, component for perceiving the suspensefulness of stories in a manner that is similar to human readers.

5.4 Evaluation 3: Modified Escape Planning

Evaluation 3 investigates how Dramatis performs when the link-cutting procedure for defining goal situations for escape planning (described in Section 3.4.5) is replaced. Specifically, rather than selecting the causal link that leads to the most easily-retrieved escape plan, thus avoiding some negative outcome, suppose an arbitrary link were selected. Would Dramatis still produce suspense ratings similar to human readers? I refer to this modified system as Dramatis-Random-Cut. By ablating this portion of the system, I can demonstrate that the intelligent selection of the link to be cut leads to a consistently correct ordering of the two versions of each story. Additionally, this ablation will demonstrate the value in finding a specific escape plan, rather than generating any escape plan. The latter strategy leads to incorrect interpretations of the story and the suspense level at that point in the story. I hypothesize that the suspense ratings produced by Dramatis-Random-Cut would widely vary from execution to execution, depending on which links were chosen, thereby giving unreliable and inconsistent results.

> **Hypothesis 3:** Dramatis-Random-Cut cannot consistently order stories according to suspense level the same way humans order stories according to suspense level.

Using this procedure for determining planning goals, it would be possible for Dramatis to determine that either version of the story is more suspenseful, rather than consistently providing the same ordering. Therefore, if this hypothesis holds, I can conclude that this algorithm is necessary in order to produce orderings of suspensefulness that match the orderings provided by human readers.

5.4.1 Method

Figure 21 shows the algorithm used by Dramatis-Random-Cut. At every time-slice, Dramatis selects one of the links that is a candidate for cutting, generates an escape

Let T be the set of time-slices given to Dramatis.

For each time-slice $t \in T$:
    Select a random causal link c to negate.
    Generate an escape plan p for c.
    Calculate the cost for p and add it to the suspense curve S.
Calculate the area under the curve for S.

Figure 21: Dramatis-Random-Cut Algorithm

plan, and calculates the suspense level. A full suspense curve is generated by randomly selecting a link at every time-slice. The results for each story pair are compared to the human ratings produced in Evaluation 1.

5.4.2 Results

The graphs below show the following: Using the link-cutting procedure above, what is the frequency, at each time-slice, that one story is rated more suspenseful than the other at that point? In the graphs below, the blue line indicates the frequency with which the link selected from the original version led to a higher suspense rating than the link selected for the alternate version, for each Time-Slice. The red line indicates when the alternate version had a higher suspense rating. The orange line shows the frequency with which both links in both stories led to maximum suspense ratings, while the green line shows the frequency of equal non-maximum suspense ratings. The gray bars show the total number of pairs of links that could be selected between the original and alternate versions of the stories.

Additionally, I report on the results of sampling and comparing 1000 possible curves, generated by randomly cutting links at each time-slice, from both versions of a story. I compare each of the 1000 curves generated from the original version to each curve generated from the alternate version of each story. I report the likelihood that the AUC (and therefore the suspense level) of a randomly generated original version curve is greater than the AUC of a randomly generated alternate version curve.

5.4.2.1 Casino Royale

Figure 22 shows the results of Evaluation 3 for *Casino Royale*. When selecting arbitrary links, it is clear (based on the blue and red lines) that Dramatis could return results where either story is judged to be more suspenseful. This is even true at the beginning of the story, before the stories diverge, when we would expect the suspense levels to be equal.

If we consider the beginning of the stories through Time-Slice 5, the stories have not yet diverged. However, Figure 22 shows that it would be possible for either story to be considered more suspenseful than the other, especially in Time-Slices 3-5. In later portions of the story, the results indicate that the original version is found to be more suspenseful in a given time-slice more frequently than the alternate version, though the possibility exists for finding the alternate version more suspenseful, most notably in Time-Slices 12-14.

When we generate 1000 random suspense curves for each version of *Casino Royale* and compare the samples, the area under the suspense curve for the original version is greater than a curve from the alternate version 92.3% of the time ($z = 32.55$, $p < .001$).

5.4.2.2 Rear Window

Figure 23 shows the results of Evaluation 3 for *Rear Window*. The blue line, indicating the frequency with which the original version is found more suspenseful, is the highest throughout the story. However, near the end of the story, there is a greater likelihood that the alternate version will be found to be more suspenseful, or that both versions will indicate maximum suspense.

Additionally, consider the space of curves generated by randomly selecting links to break. When I sample 1000 randomly generated curves from both versions of *Rear Window*, the area under the curve for the original version is greater than a curve from

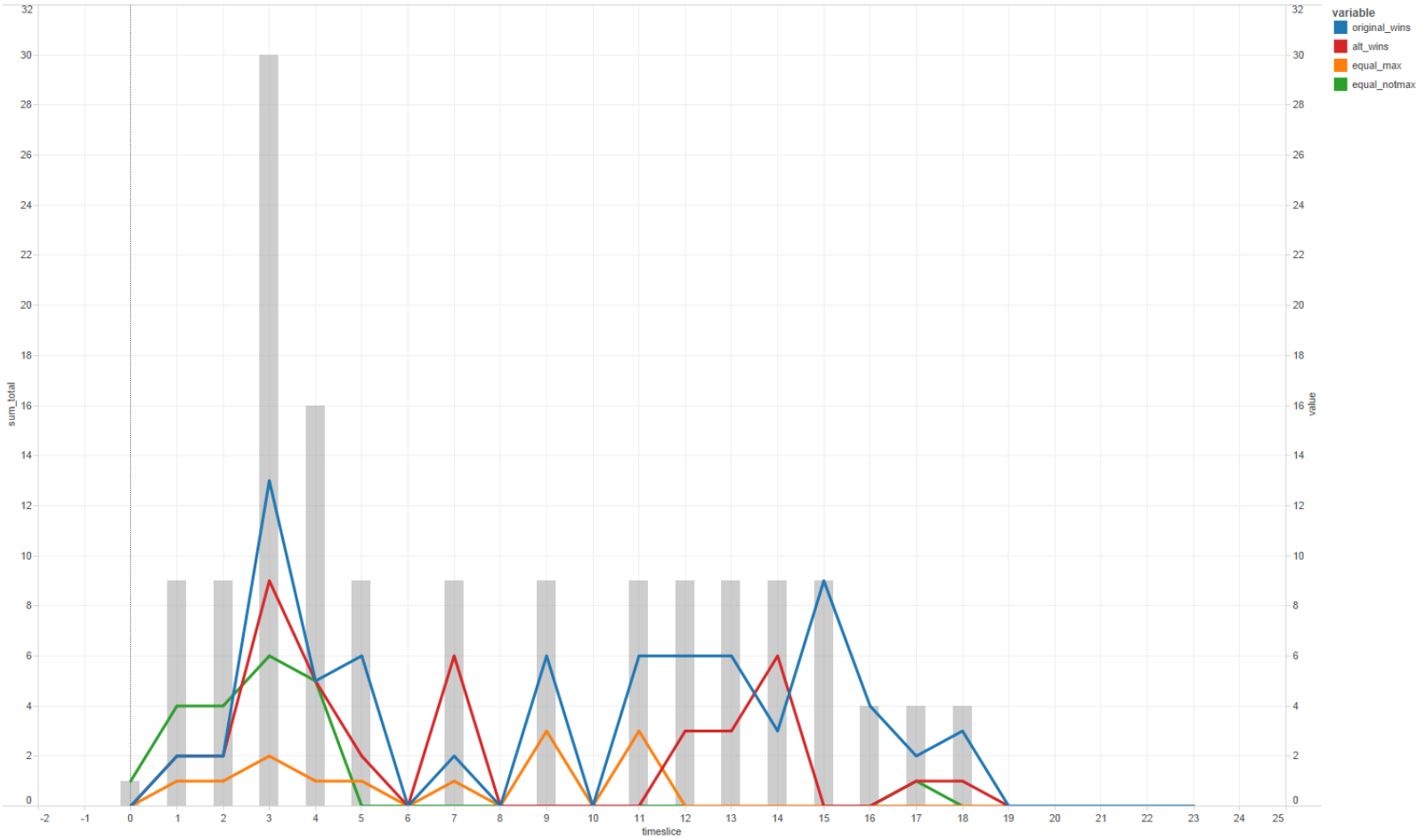

Figure 22: Distribution of suspense ratings for *Casino Royale* in Evaluation 3

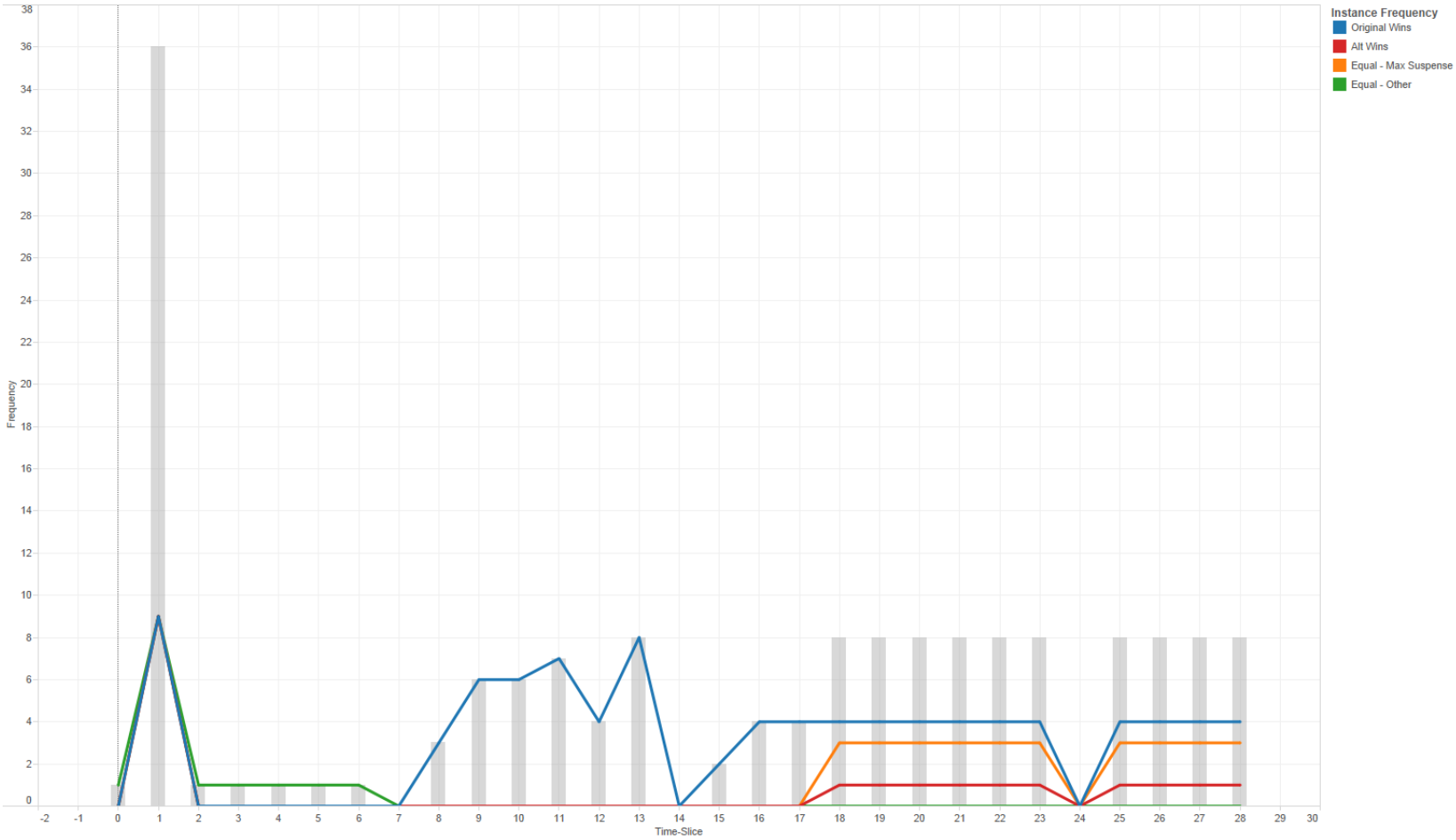

Figure 23: Distribution of suspense ratings for *Rear Window* in Evaluation 3

the alternate version 99.8% of the time ($z = 38.59$, $p < .001$).

We can look at the results by time-slice (Figure 23) in greater detail by dividing the story into thirds. In the first third (through Time-Slice 7), the results indicate that the suspense levels most often would be equal, with some chance that either version would be found more suspenseful. During this section of the story, before the stories have diverged, we expect that the suspense ratings are equal. In Time-Slices 8-17, Evaluation 3 indicates that the original version would always be found to be more suspenseful, regardless of which links were cut. Recall, however, from Study 1, that the difference in suspense for most of this portion of the story was marginal. The suspensefulness of the original version did not substantially increase until Time-Slice 16. However, in the final third (starting with Time-Slice 18), Evaluation 3 shows that it is possible for the alternate version to be found to be more suspenseful when cutting arbitrary causal links. According to the Dramatis results in Evaluation 1, this is precisely the time that we expect the original version to be more suspenseful. Thus, the arbitrary link-cutting procedure produces inconsistent results in the first and last third of the story, while indicating that the original version will always be more suspenseful in the middle third.

5.4.2.3 Harry Potter and the Half-Blood Prince

Finally, Figure 24 shows the results of the *Harry Potter* stories in Evaluation 3. For the first ten Time-Slices, while there is a strong chance that the two links would lead to equal suspense ratings, arbitrary selection could lead to a preference for either of the two stories. For the remainder of the story, however, we see no chance for equal ratings, but several situations where either story could be found to be more suspenseful, depending on which links were selected.

When looking at a sample of randomly generated curves, the area under the curve for the original version is greater than the area underneath an alternate version curve

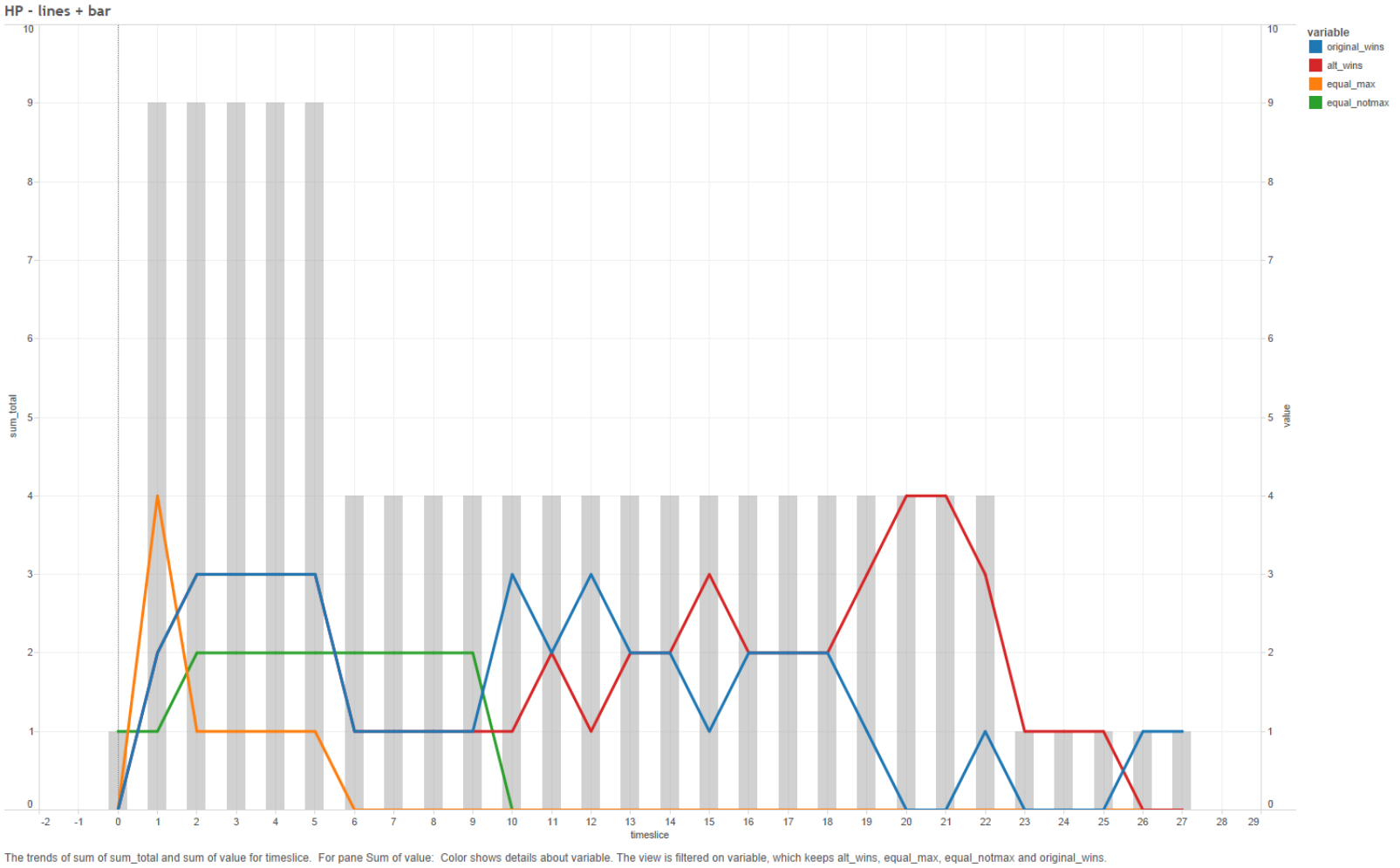

Figure 24: Distribution of suspense ratings for *Harry Potter* in Evaluation 3

62.9% of the time ($z = 10.01$, $p < .001$).

5.4.3 Discussion

The results of sampling from the space of possible random curves demonstrates that comparing large samples of curves for two versions of a story and taking the "majority vote" between the original and alternate versions will produce the same suspense ratings as human readers or the full Dramatis system. Thus a random sampling search algorithm could be devised that selected the more suspenseful version with between 62% and 99% accuracy, depending on the particulars of the domain. Therefore, my algorithm for deterministically selecting the causal link to cut to generate the lowest-cost escape plan is sufficient for generating accurate suspense ratings, but not the only algorithm that can exist. However, note that the sampling strategy overwhelmingly selected the original version of *Rear Window* as the more suspenseful version, while human readers did not reach complete consensus about that decision. This may suggest that there is a weakness in the sampling strategy, or simply that looking at the margin of the sampling decision does not indicate the degree of agreement with people. Additionally, random sampling tends to overestimate the suspense level. Given that Dramatis always produces the escape plan with minimum cost, any suspense value produced by Dramatis-Random-Cut will, by definition, be equal to or greater than the value produced by Dramatis. This tendency to over-estimate may explain the circumstances where Dramatis-Random-Cut considers the alternate version to be more suspenseful.

Perhaps the biggest flaw with selecting arbitrary links comes at the beginning of the stories. Before the stories diverge, we would expect that the stories have equal suspense ratings. While this study shows that this is possible, it is also possible that one story could be deemed more suspenseful than the other. This simply makes no sense when there is no difference between the stories at that point.

When looking at the data broken down by time-slice, the data suggests that there are areas where we expect one version of a story to be more suspenseful but have a distinct possibility of being told that the other version is more suspenseful. For example, Dramatis-Random-Cut generally favors the original version of *Rear Window*, but the times where the alternate version could be favored are precisely the times when we would expect it to be less suspenseful. Despite these intuitions about which portion of a story should be more suspenseful, we cannot conclusively say whether these results conflict with human responses without collecting more data from human readers with ratings at multiple points throughout the story.

Hypothesis 3 is partially supported by the results of this evaluation. These results show that the intelligent link-cutting algorithm is sufficient for Dramatis to produce suspense ratings similar to human readers. Additionally, I have shown that the random link-cutting strategy selects the same ordering as human readers between 62-99% of the time, depending on the domain. Thus, there is some likelihood of the algorithm not producing the right ordering, albeit quite low in certain domains. The random-link cutting algorithm has one significant flaw, which is that the random selection can lead to identical stories (or portions of stories) being given different ratings.

5.5 Conclusions

The three evaluations described in this chapter demonstrate the following about Dramatis:

- When stories are altered to become less suspenseful using techniques indicated by my definition of suspense, Dramatis produces correspondingly lower ratings.
- The suspense ratings produced by Dramatis are largely consistent with those of human readers.
- The MEI-P Situtation Model is a sufficient component of Dramatis. Without it,

Dramatis produces ratings that do not consistently correspond to my definition of suspense, nor to ratings provided by human readers.

- The algorithm for generating escape plan goal situations by selecting causal links to cut is sufficient, but not necessary, for Dramatis to produce ratings similar to those provided by human readers. Sampling a large space of curves generated by choosing causal links randomly may also be an effective strategy for calculating suspense.

In Evaluation 1, I demonstrated that Dramatis finds the original version of each story pair to be more suspenseful than the alternate versions. Recall that I created the alternate versions with the intent that they would be less suspenseful, making changes that correspond to my reinterpretation of Gerrig and Bernardo's definition of suspense. Additionally, human readers selected the original version more frequently for all three pairs, with statistically significant preferences in the case of *Casino Royale* and *Harry Potter*. Further, participants were asked to rate the suspensefulness of all six stories. In each pair, particpants found the original version to be significantly more suspenseful than the alternate version.

Evaluation 2 was an ablative study, wherein the MEI-P Situation Model was removed from the calculation of planning operator costs. I hypothesized that Dramatis would find that, within pairs, the stories were either indistinguishable, or the alternate version was more suspenseful. For *Casino Royale*, the original version remained more suspenseful, largely due to one specific change between the versions. The ablated Dramatis was unable to find any difference in suspense between the two versions of *Rear Window*. Finally, without the MEI-P model, Dramatis continued to find the original version of *Harry Potter* to be more suspenseful than the alternate version.

Evaluation 3 investigated the consequences of replacing intelligent causal link selection with arbitrary selection when identifying goals for escape planning. I expected

that this arbitrary selection would lead to results that varied across executions. However, when looking at the space of suspense curves that could be generated using this method, we found that taking a large sample from that space would produce the same results as the full Dramatis system. When looking at results by individual time-slice, there is reason to believe that this strategy could produce incorrect results about which story is more suspenseful in a given moment. Confirming this result requires collection of more detailed data from human readers.

These studies demonstrate that the ratings produced by the Dramatis suspense model match human suspense ratings, and the model components are sufficient for its success.

CHAPTER VI

CONCLUSIONS

The principal contributions of this work are an empirically evaluated computational model of suspense derived from Gerrig and Bernardo's psychological model, and a methodology for domain and knowledge engineering from natural language data based in qualitative methods. The suspense model introduced here is applicable to computational creativity research, notably story generation and interactive narrative work. Additionally, the qualitative knowledge engineering methodology is applicable to any knowledge-intensive system that looks to represent natural language data collected from human subjects.

6.1 Summary

The primary contribution of this work is a computational model of suspense from the audience perspective. The model is based on Gerrig and Bernardo's definition of suspense, which states that readers feel suspense when led to believe that the quantity or quality of paths for a hero's escape from a negative outcome is decreasing. I reformulate this psychological approach to suspense as a step towards creating a computational representation of the model. Given the belief that a character can face a negative outcome, one can assume that this outcome will occur and search, on behalf of the character, for the single most likely plan in which the protagonist avoids this outcome. I measure the quality of such a plan as its perceived likelihood of success, from the perspective of the audience. As the likelihood of success decreases, the level of suspense is expected to increase.

Dramatis reads in stories in a symbolic-logic format. Using either by the details

of the story or its knowledge of appropriate genres, Dramatis assumes that some impending failure for the protagonist (such as falling unconscious) will occur. Dramatis uses its knowledge of genre, in the form of scripts, to identify what causal links can be broken in an effort to prevent this negative outcome from occurring. The suspense model generates an escape plan on behalf of the protagonist that breaks one of the causal links in the script, thereby making it impossible for the negative outcome to happen.

As Dramatis reads the story, it adds each event and story element to a model of reader memory. This memory model uses a spreading activation network to represent the relatedness of story elements, where a higher activation indicates greater salience in the reader's mind. In order to generate the escape plan that a reader considers to be most likely to succeed, Dramatis uses the activation of nodes in the salience model to inform the costs of generating the escape plan. As the overall cost of the escape plan increases, the level of suspense in that moment also increases.

In the first of three evaluations, I demonstrated that Dramatis provides suspense ratings comparable to human readers. When presented with two versions of stories, human readers and Dramatis agreed on which version of the story was more suspenseful. In the second evaluation, I ablated the function of the reader salience model in the escape planning process. That is, plan operators were given pre-defined equal values rather than being influenced by the salience model. When given the same set of stories as before, the modified system could not reliably match the human-provided suspense ratings. In the final study, I replaced the algorithm for identifying which causal link to cut during escape planning with a random selection process. I found that sampling from the space of randomly generated suspense curves produces the same results as the full Dramatis system. This indicates that sampling may be an effective approach to comparing the suspensefulness of stories.

Additionally, in this dissertation, I have introduced an approach to knowledge engineering based on qualitative methods. As Dramatis required background knowledge in the form of scripts and planning operators, I developed this method for knowledge acquisition from a natural language corpus that mitigates engineer bias. In this methodology, domain engineers collect a natural language corpus and use an iterative process of coding to identify the sequences of actions being described. The multiple iterations allow domain engineers to refine codes and be certain that entries are being properly included or excluded from the resulting knowledge set.

6.2 Limitations

Dramatis represents a model of suspense for a subset of the space of suspenseful stories. In Chapter 2, I categorized the methods used to produce suspense in a number of popular films. I claim that Dramatis can only model a type of suspense that I refer to as *procedural expectation-based suspense*, or suspense that relies on audience knowledge of genre actions or characters' plans. I intentionally exclude *outcome expectation suspense*, a form of suspense where an outcome is expected or desired by the audience, but the specifics of how that outcome could come to pass are not known in detail. While an audience might have an abstract concept of how the outcome could be achieved, there is not enough information for the audience to consider the causal relationships of the possible events. It is likely that the audience feels suspense, but the only active cognitive processes that can occur is "wait and see what unfolds." While this may anecdotally reflect our own experiences with certain suspenseful stories, there is little research on the cognitive mechanisms that occur in these situations. Further research is necessary to capably model this other type of suspense. Additionally, there may be other categories of suspense that I have not outlined in this work.

Dramatis can be used only to compare variations of the same story, but it cannot

be used to compare radically different stories. Comparing significantly different stories reliably would require significantly more domain knowledge. Currently, because of how the procedural knowledge was generated for Dramatis, there is no meaningful comparison. For an effective comparison, Dramatis would need access to a wide array of scripts and plan operators. However, this is not simply a matter of combining the scripts and operators from the three domains used in the evaluations of Dramatis and generating suspense curves again. While Dramatis would have access to the combined knowledge, but the libraries would still be disjoint domains. For a cross-story comparison to be valid, we would need some intersection in the knowledge for the domains in question. If the operators and scripts were designed to interact with each other, rather than an separate domains, then there would be room for comparison. This type of combined knowledge would guarantee that the granularity of operators and script events, for example, were at the same depth. Additionally, rather than producing the corpus or domain knowledge in isolated prompts for each domain, we would want to combine the prompts, so as to get a greater sense of the interaction in domains. Such a process for knowledge acquisition and engineering would need further assessment, but could be based on the methodology presented in Chapter 4. Each of these factors would need to be controlled for in order for any cross-story comparison to be valid.

The stories used to evaluate Dramatis may be considered relatively simplistic examples of suspense. In each story, the protagonist deals with one immediate negative consequence, and Dramatis did not consider the larger goals of the characters and the story (e.g., In *Casino Royale*, James Bond was trying to prevent a terrorist from earning money). Many stories have these layers of goals and consequences at work, and as a result, suspense can come from immediate negative consequences (e.g., being poisoned) or long-term consequences (e.g. funding terrorism). Therefore, expansions to Dramatis may wish to consider how to handle these layered goals and suspenseful

situations, and how the suspense from long-term goals influences or interacts with the suspense from more immediate negative outcomes.

While Dramatis reports a numerical level of suspense for each time-slice, there is little that I can say with respect to the meaning of these numbers. The value that Dramatis outputs is only useful in comparison to points in the same story or between similar stories. Because we have no human data for the suspense levels mid-story, I cannot say anything about whether human readers experience the peaks and troughs in suspense predicted by Dramatis. Further research could address this lack of human data, which may allow me to attach greater meaning to the numerical suspense levels.

While describing the various definitions of suspense in Chapter 2, I noted that character affinity is one of the key components of several definitions. However, Gerrig and Bernardo's definition does not consider character affinity, nor does my reformulation of that definition. It is possible that character affinity could be represented as a boolean value or a scalar on a [-1, 1] scale. Intuitively, humans, and therefore Dramatis, would feel different levels of suspense for characters for whom they have different levels of affinity, including characters that are strongly disliked by human readers. Dramatis currently has no way to measure character affinity. Additionally, there is a level of bias to be accounted for when asking human participants about affinity for characters. In particular, the stories used during evaluation were all adapted from popular books and films. Many of my participants likely already had preconceived affinity for James Bond or Harry Potter, for example. In contrast, anecdotal evidence suggests that it would be difficult to generate new short stories with unknown characters that will be liked by a human audience. Inclusion of character affinity in Dramatis should be preceded by further study into what makes an audience like or dislike characters in the first place.

6.3 Toward Generation of Suspenseful Stories

In this section, I discuss the potential applications of Dramatis to story generation. As a human behavior model of audience aesthetics, Dramatis provides an opportunity for story generation systems to reason about and evaluate stories according to the suspense felt by the audience. With access to Dramatis, story generation researchers can address the calls of Zagalo [78] and Szilas [71] to consider the emotions of the audience when generating stories. Specifically, by intentionally generating suspenseful stories, story generation systems can be expected to produce works that readers find more entertaining [13, 72].

In Section 2.8, I described several systems that model aesthetics as part of story generation. Many of these systems use pre-defined models of tension or an ideal sequence of tension levels as guides in the generation process. However, these models are based solely on the author or designer's perception of what makes for a good story. They assume that the audience uses the same metrics for enjoyment, rather than explicitly considering the emotions or enjoyment of their audiences. Dramatis provides an opportunity to move beyond these strategies to story generation. As an audience model of suspense, Dramatis affords a story generation system with the capacity to focus on the audience's perception of the story, rather than the author's goals. However, some sense of the author's intention will remain necessary in a story generation system employing Dramatis. Because Dramatis does not define an ideal level of suspense, or any notion of story quality beyond suspense, authors and designers may need to determine for themselves what patterns of suspense are most appropriate.

In its simplest form, the suspense ratings produced by Dramatis could be applied as a heuristic or evaluation function, leading a story generation system towards stories that are more suspenseful. Such story generation systems could use this heuristic

to evaluate a space of stories, potentially in various states of completion, and determine which story is most suspenseful. A designer may also choose not to search for the most suspenseful story, but instead achieve some minimal threshold of suspense for the story-in-progress. Similarly, an interactive narrative system could produce a set of possible paths that the user could take. Dramatis could evaluate these paths, generating the suspense curve for each of them. The story generation system could compare the overall suspense levels indicated by the curves, or it could compare the curves to an author- or designer-specified ideal suspense curve. A drama manager could then attempt to guide a player through a suspenseful version of the narrative. While this generate-and-check method is simple, it would achieve the goals of improving audience enjoyment in stories by presenting more suspenseful stories than previously generated.

It may also be possible for a story generation system to reason about the escape plans generated by Dramatis. Consider a situation in which Dramatis is given an incomplete story to evaluate. The model evaluates the story, rates the suspense, and produces the corresponding escape plan (if a dire consequence has been detected). The story generation system could then reason about new events to add to the story to increase or decrease the suspense of the story-so-far. Suspense would be increased by inserting events that reduce the viability of the escape plan, while decreased suspense could come from adding events to increase the escape plan's likelihood of success. Depending on the exact nature of the story planner, it may be possible to insert or remove events from anywhere in the story, rather than just working from the end [47, 36]. This process could continue iteratively, until some threshold of minimal suspense was achieved. The story generation system would then have to design some means of resolving the suspense and bringing the story to a close.

One could also incorporate the escape plans more directly into the stories being generated. Recall the example used by Gerrig and Bernardo from Ian Fleming's

Casino Royale novel. In the example, Fleming places James Bond in a situation where he is in danger of being shot. The story then shifts to Bond's inner monologue as he contemplates ways that he could escape this fate. Fleming produces (from Bond's perspective) a possible solution—getting the casino staff's attention—and then removes it by informing us that they are not paying attention. Fleming produces a second solution—getting help from Bond's friends—and removes this option as well. Fleming thus manipulates the audience's suspense level by (a) reducing the quantity of escape plans available to James Bond, and (b) specifically calling out escape plans that are likely to be salient in the reader's memory and reducing their likelihood of success. A story generation system could use the escape plans produced by Dramatis for a similar effect, particularly with a small modification to Dramatis to produce a set of escape plans rather than a single plan. The story generation system could specifically incorporate the escape plans into the story and change the state of the story so that these plans are no longer available. Each plan produced and revoked would increase audience suspense by reducing the quantity of plans, and more importantly, removing those plans that are most salient in memory model. Any remaining escape plans will likely have a much higher retrieval cost, thereby increasing the suspensefulness of the story.

Each potential story generation process that I have proposed thus far is computationally expensive. Story generation is already **PSPACE**, and using Dramatis as an evaluation function adds an additional **PSPACE** process to the generation task. That is, reasoning about escape plans in the generation process, either by using adversarial planning to increase suspense or by introducing and revoking possible solutions, adds to the complexity. The former strategy necessitates using Dramatis to evaluate a story-in-progress, altering the story to manipulate the suspense level, and re-evaluating the new story to confirm the expected effects on the story, thereby inserting additional **PSPACE** planning to the generation procedure. The latter

strategy requires Dramatis to generate multiple escape plans at each time-slice and introduce them to the reader, thereby changing the knowledge available to the reader and manipulating the suspense level. Determining where in the story to apply these methods will require further searching. While these strategies may potentially produce more suspenseful and more entertaining stories, there is certainly an additional complexity cost to consider.

Given an existing or previously generated story, Dramatis could be employed to determine the most suspenseful sjuzhet. The process used to determine the most suspenseful sjuzhet could be similar to the procedures used by Suspenser [19] or Prevoyant [5], which sought to find the most suspenseful and surprising versions of a story, respectively. A system could attempt to add, delete, or reorder events with the intent to produce a more suspenseful telling of the story, with Dramatis evaluating the consequences of making the change. Any change to the story would effect the salience of elements in the memory model, thereby changing the set of escape plans available and their costs. Similarly, a system could be devised so that a human author, rather than an intelligent system, could alter or delete events and see the resultant changes to the suspense curve. In this manner, Dramatis would be employed as a critic for the author's story rather than serving directly as part of a story generation system.

Using any one of these methods, Dramatis augments current approaches to story generation. Each approach described in this section potentially provides the ability to generate suspenseful stories, which are likely to be considered more enjoyable than the stories that are currently generated. Additionally, each approach emphasizes the use of suspense from the audience perspective, rather than a pre-defined ideal notion of suspense from the author or designer. In the future, it may be valuable to compare these approaches and determine which has the greatest effect at producing suspenseful or enjoyable stories. The results of such a study may be dependent on the story domain, the knowledge provided to the story generator and Dramatis, and

the intentions of the author or designer.

6.4 Future Work

In addition to the applications to story generation research, this work provides several avenues for continuing work. In this section, I consider how mid-story suspense ratings from Dramatis could be compared to human suspense ratings, as well as the value of allowing Dramatis to reason about multiple escape plans. Finally, I describe possible work expanding the qualitative knowledge engineering methodology.

6.4.1 Within-Story Suspense

Dramatis ratings have only been compared to human ratings for the suspensefulness of an entire story. Further research is required to determine if Dramatis suspense ratings correlate with humans' ratings on a time-slice by time-slice basis. That is, do the suspense curves generated by Dramatis resemble the curves we would see from human readers if they were asked to provide ratings at multiple points in the story? Asking readers to draw their own suspense curve while reading or after reading may provide some detail about the suspense levels at pre-determined points in the story. The feasibility of such a post-hoc interview protocol would need to be assessed.

6.4.2 Multiple Escape Plans

Currently, my reformulation of Gerrig and Bernardo's suspense model works, in part, from the assumption that humans are resource-bounded and incapable of generating the entire search space when problem-solving on behalf of the protagonist. As a consequence, Dramatis only generates one escape plan for the protagonist, which does not necessarily require the generation of the entire search space. However, it may be possible for humans to generate multiple escape plans, particularly if similar plans exist, or if several plans are sufficiently salient in memory. Further study is required in order to determine the extent to which multiple escape plans are available

to human readers. It may be possible to ask readers to propose solutions for characters in a limited amount of time. Such a study would provide insight into the escape plan process as well as the memory model. Modification of Dramatis to reason about multiple escape plans may also provide insight into how reader memory influences the suspense, while possibly allowing for better story generation. Altering Dramatis to discover and track multiple escape plans requires only small changes to the planning algorithm. The planning search may have to go deeper in the search tree, although setting a threshold of salience should mitigate this issue. When tracking escape plans, Dramatis will have to compare newly observed events to a set of plans rather than a single plan. Depending on the results of studying human readers, it may be necessary for Dramatis to conduct the planning task again to see if any new plans are salient, even if the previous escape plans are still active.

6.4.3 Qualitative Knowledge Engineering

Another direction for future work is the expansion of the qualitative knowledge engineering methodology. While I was able to collect sets of actions from a natural language corpus, converting these actions to planning operators required the creation of preconditions and effects which did not come directly from the corpus. If the methodology could be expanded to provide information about the causality of events, it would improve the knowledge engineering of both the operators and the scripts being generated. Similarly, it would be useful to glean information about intentionality from this methodology. Finally, one might be interested in determining what types of corpora work best with this methodology. Survey and interview responses, along with data-mined natural language corpora, should work well. It may be possible to use corpora from other media, such as generating scripts or operators directly from television, film, or games. Further study is required to see what changes to the methodology would be necessary for these other media.

6.5 Concluding Remarks

As a computational model of suspense, validated against human suspense ratings, Dramatis serves as a means of improving the stories produced by story generation and interactive narrative systems. By modeling human responses to a creative artifact, rather than guiding the creation process through some authorial model of aesthetic, there is hope to create products that will be appreciated by human readers and players.

The value of story structure and emotional involvement has long been known to dramatists and entertainers. However, the attempts at these features made by computational systems have incorrectly focused on the notion of an ideal tension curve or dramatic arc, thus guiding stories according to the author or designer's notion of what made for a "good" story, without considering the impact on the audience. Expert storytellers craft narratives for entertainment purposes. If computational story generation systems fail to tell good stories, then they fail to entertain. A computationally creative system must entertain if it is to be comparable to human experts.

Narratologists have demonstrated that suspense in stories contributes to reader enjoyment. By incorporating a model of suspense into computationally creative systems, such as story generators or interactive narrative systems, it will be possible to generate suspenseful stories intentionally. The stories created in concert with this model will, simply by virtue of being suspenseful, have a higher capacity for entertainment than those created without any model of audience aesthetic.

As our ability to model the emotions and aesthetics of humans improves, so will our ability to intelligently generate creative artifacts that entertain and affect people. The cognitive processes of humans are centered around our ability to understand narratives. Thus, any artificial intelligence system that hopes to fully model humans must be able to not only generate narratives, but also appreciate narratives in a

human-like manner. Therefore, this work not only builds toward the improvement of computationally creative systems, but also toward our ability to model the greater cognitive processes of humans.

APPENDIX A

KNOWLEDGE STRUCTURES

This appendix contains the planning operators and scripts created from the knowledge engineering methodology and used in the evaluation of Dramatis.

A.1 Operators

The operators presented here are represented as follows. The first line (preceded by `op`) gives the operator name and parameters. The second line gives the constraints (`con`). The last three lines give the preconditions (`prec`), add effects (`add`), and delete effects (`del`), respectively.

Constraints are a special subset of the operator's preconditions. These preconditions establish immutable facts about the parameters—for example, that a parameter variable represents a person or place. At least one proposition must be given in the constraints for each parameter. A constraint beginning with `neq` establishes that the two variables following it cannot be bound to the same symbol (i.e., the `go` operator requires that the origin and destination locations are different). No proposition in the constraints is eligible to be part of a causal link being cut by Dramatis during escape plan generation.

A.1.1 Casino Royale

```
op: (apply-antidote ?actor ?antidote ?target)
    con: (person ?actor) (person ?target) (antidote ?antidote)
    prec: (has ?actor ?antidote) (poisoned ?target) (sick ?target)
    add: (healthy ?target)
    del: (poisoned ?target) (has ?actor ?antidote) (sick ?target)

op: (approach ?actor ?target ?location ?from)
    con: (person ?actor) (person ?target) (place ?location)
```

```
          (place ?from) (neq ?actor ?target) (neq ?location ?from)
    prec: (at ?target ?location) (at ?actor ?from)
    add: (at ?actor ?location)
    del: (at ?actor ?from)

op: (argue ?personA ?personB)
    con: (person ?personA) (person ?personB) (neq ?personA ?personB)
    prec:
    add:
    del:

op: (attach-leads ?actor ?target ?defib ?location)
    con: (person ?actor) (person ?target) (defibrillator ?defib)
         (place ?location)
    prec: (at ?actor ?location) (at ?target ?location)
          (at ?defib ?location) (conscious ?actor)
    add: (leads-attached ?defib ?target)
    del:

op: (attend-to ?actor ?food ?place)
    con: (person ?actor) (edible ?food) (place ?place)
    prec: (unattended ?food) (at ?actor ?place) (at ?food ?place)
          (belongs-to ?food ?actor)
    add: (has ?actor ?food)
    del: (unattended ?food) (at ?food ?place)
         (belongs-to ?food ?actor)

op: (avoid-punch ?actor)
    con: (person ?actor)
    prec: (punched ?actor)
    add:
    del: (punched ?actor)

op: (become-sick ?actor)
    con: (person ?actor)
    prec: (poisoned ?actor) (healthy ?actor)
    add: (sick ?actor) (dying ?actor)
    del: (healthy ?actor)

op: (bribe ?actor ?target ?money)
    con: (person ?actor) (person ?target) (money ?money)
         (neq ?actor ?target)
    prec: (has ?actor ?money) (evil ?actor) (neutral ?target)
    add: (has ?target ?money) (evil ?target)
    del: (has ?actor ?money) (neutral ?target)
```

```
op: (bump ?actor ?bumped)
    con: (person ?actor) (person ?bumped) (neq ?actor ?bumped)
    prec:
    add:
    del:

op: (choke ?actor)
    con: (person ?actor)
    prec: (poisoned ?actor)
    add: (sick ?actor)
    del:

op: (confront ?actor ?target)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
    prec:
    add: (frightened ?target)
    del:

op: (contact ?speakerA ?speakerB)
    con: (person ?speakerA) (person ?speakerB)
         (neq ?speakerA ?speakerB)
    prec:
    add:
    del:

op: (defibrillate ?actor ?defib ?target)
    con: (person ?actor) (person ?target) (defibrillator ?defib)
    prec: (dying ?target) (leads-attached ?defib ?target)
          (conscious ?actor)
    add: (wakeable ?target)
    del: (dying ?target)

op: (deliver-food ?waiter ?food ?customer)
    con: (person ?waiter) (person ?customer) (edible ?food)
         (neq ?waiter ?customer)
    prec: (has ?waiter ?food) (ordered ?customer ?food)
          (waiter ?waiter)
    add: (has ?customer ?food)
    del: (has ?waiter ?food) (ordered ?customer ?food)

op: (die ?victim)
    con: (person ?victim)
    prec: (dying ?victim) (unconscious ?victim)
    add: (dead ?victim)
```

```
    del: (alive ?victim)

op: (disguise ?person ?costume)
    con: (person ?person) (costume ?costume)
    prec: (has ?person ?costume) (hidden ?person)
    add: (waiter ?person)
    del: (hidden ?person)

op: (drink ?actor ?drink)
    con: (person ?actor) (drink ?drink)
    prec: (has ?actor ?drink)
    add:
    del:

op: (drink-poisoned ?actor ?drink)
    con: (person ?actor) (drink ?drink)
    prec: (has ?actor ?drink) (poisoned ?drink)
    add: (poisoned ?actor)
    del:

op: (evil-smile ?actor)
    con: (person ?actor)
    prec:
    add:
    del:

op: (extort ?actor ?target ?item)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
         (object ?item)
    prec: (has ?actor ?item)
    add:
    del:

op: (fake-death ?actor)
    con: (person ?actor)
    prec: (sick ?actor)
    add: (appears-dead ?actor)
    del:

op: (fall ?actor)
    con: (person ?actor)
    prec:
    add:
    del:
```

```
op: (give ?actor ?item ?target)
    con: (person ?actor) (person ?target) (object ?item)
         (neq ?actor ?target)
    prec: (has ?actor ?item)
    add: (has ?target ?item)
    del: (has ?actor ?item)

op: (go ?actor ?from ?to)
    con: (person ?actor) (place ?from) (place ?to) (neq ?from ?to)
    prec: (at ?actor ?from)
    add: (at ?actor ?to)
    del: (at ?actor ?from)

op: (go-threatened ?actor ?from ?to)
    con: (person ?actor) (place ?from) (place ?to) (neq ?from ?to)
    prec: (at ?actor ?from) (threatened ?actor)
    add: (at ?actor ?to)
    del: (at ?actor ?from)

op: (grab ?actor ?item ?from)
    con: (person ?actor) (object ?item) (place ?from)
    prec: (at ?item ?from) (at ?actor ?from)
    add: (has ?actor ?item)
    del: (at ?item ?from)

op: (hide ?actor)
    con: (person ?actor)
    prec:
    add: (hidden ?actor)
    del:

op: (kneel ?actor)
    con: (person ?actor)
    prec:
    add:
    del:

op: (leap ?actor)
    con: (person ?actor)
    prec:
    add:
    del:

op: (leave-unattended ?actor ?food ?place)
    con: (person ?actor) (edible ?food) (place ?place)
```

```
    prec: (has ?actor ?food) (at ?actor ?place)
    add: (unattended ?food) (at ?food ?place)
         (belongs-to ?food ?actor)
    del: (has ?actor ?food)

op: (lick-lips ?actor)
    con: (person ?actor)
    prec:
    add:
    del:

op: (look-for-person ?actor ?target)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
    prec:
    add:
    del:

op: (make-antidote ?actor ?antidote)
    con: (person ?actor) (antidote ?antidote)
    prec: (smart ?actor)
    add: (has ?actor ?antidote)
    del:

op: (make-drink ?actor ?drink ?someone)
    con: (person ?actor) (person ?someone) (drink ?drink)
         (neq ?actor ?someone)
    prec: (waiter ?actor) (ordered ?someone ?drink)
    add: (has ?actor ?drink)
    del:

op: (offer-antidote ?actor ?antidote ?target)
    con: (person ?actor) (antidote ?antidote) (person ?target)
         (neq ?actor ?target)
    prec: (evil ?actor) (poisoned ?target) (has ?actor ?antidote)
    add:
    del:

op: (offer-seat ?actor ?target)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
    prec:
    add:
    del:

op: (open-bottle ?actor ?bottle)
    con: (person ?actor) (bottle ?bottle)
```

```
    prec: (has ?actor ?bottle)
    add: (open ?bottle) (drink ?bottle)
    del:

op: (order ?actor ?waiter ?food)
    con: (person ?actor) (person ?waiter) (edible ?food)
         (neq ?actor ?waiter)
    prec: (waiter ?waiter) (notmade ?food)
    add: (ordered ?actor ?food)
    del: (notmade ?food)

op: (order-drink ?actor ?waiter ?drink)
    con: (person ?actor) (person ?waiter) (drink ?drink)
         (neq ?actor ?waiter)
    prec: (waiter ?waiter)
    add: (ordered ?drink)
    del:

op: (pass-out ?actor)
    con: (person ?actor)
    prec: (sick ?actor) (conscious ?actor)
    add: (unconscious ?actor)
    del: (conscious ?actor)

op: (pickpocket ?actor ?target ?item)
    con: (person ?actor) (person ?target) (object ?item)
         (neq ?actor ?target)
    prec: (has ?target ?item)
    add: (has ?actor ?item)
    del: (has ?target ?item)

op: (poison-drink ?actor ?drink ?poison)
    con: (person ?actor) (drink ?drink) (poisonous ?poison)
    prec: (has ?actor ?drink) (has ?actor ?poison) (evil ?actor)
    add: (poisoned ?drink)
    del: (has ?actor ?poison)

op: (poison-drink-distracted ?actor ?drink ?poison)
    con: (person ?actor) (drink ?drink) (poisonous ?poison)
    prec: (unattended ?drink) (has ?actor ?poison) (evil ?actor)
    add: (poisoned ?drink)
    del: (has ?actor ?poison)

op: (procure-antidote ?actor ?owner ?antidote ?cash)
    con: (person ?actor) (person ?owner) (antidote ?antidote)
```

```
          (money ?cash) (neq ?actor ?owner)
    prec: (has ?actor ?cash) (has ?owner ?antidote)
    add: (has ?actor ?antidote) (has ?owner ?cash)
    del: (has ?actor ?cash) (has ?owner ?antidote)

op: (punch ?actor ?target)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
    prec:
    add: (punched ?target)
    del:

op: (read ?actor ?item)
    con: (person ?actor) (object ?item)
    prec: (has ?actor ?item)
    add:
    del:

op: (rub-chest ?actor)
    con: (person ?actor)
    prec:
    add:
    del:

op: (salute-drink ?actor ?target ?drink)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
         (drink ?drink)
    prec: (has ?actor ?drink)
    add:
    del:

op: (search ?actor)
    con: (person ?actor)
    prec:
    add:
    del:

op: (send-signal ?sender ?receiver)
    con: (person ?sender) (person ?receiver) (neq ?sender ?receiver)
    prec:
    add: (got-signal ?sender) (evil ?receiver)
    del:

op: (sit ?actor)
    con: (person ?actor)
    prec:
```

```
    add:
    del:

op: (smile ?actor)
    con: (person ?actor)
    prec:
    add:
    del:

op: (swap ?personA ?personB ?itemA ?itemB)
    con: (person ?personA) (person ?personB) (neq ?personA ?personB)
         (object ?itemA) (object ?itemB) (neq ?itemA ?itemB)
    prec: (has ?personA ?itemA) (has ?personB ?itemB)
    add: (has ?personA ?itemB) (has ?personB ?itemA)
    del: (has ?personA ?itemA) (has ?personB ?itemB)

op: (swirl ?actor ?drink)
    con: (person ?actor) (drink ?drink)
    prec: (has ?actor ?drink)
    add:
    del:

op: (take-punch ?victim)
    con: (person ?victim)
    prec: (punched ?victim)
    add: (unconscious ?victim)
    del: (punched ?victim)

op: (talk-b ?speakerA ?speakerB)
    con: (person ?speakerA) (person ?speakerB)
         (neq ?speakerA ?speakerB)
    prec:
    add:
    del:

op: (talk ?speakerA ?speakerB ?location)
    con: (person ?speakerA) (person ?speakerB) (place ?location)
         (neq ?speakerA ?speakerB)
    prec: (at ?speakerA ?location) (at ?speakerB ?location)
    add: (talking ?speakerA ?speakerB) (talking ?speakerB ?speakerA)
    del:

op: (threaten ?actor ?target ?weapon)
    con: (person ?actor) (person ?target) (weapon ?weapon)
         (neq ?actor ?target)
```

```
    prec: (has ?actor ?weapon) (evil ?actor)
    add: (threatened ?target)
    del:

op: (twirl-vial ?actor ?antidote)
    con: (person ?actor) (antidote ?antidote)
    prec: (has ?actor ?antidote)
    add:
    del:

op: (vomit ?actor ?place ?toilet)
    con: (person ?actor) (place ?place) (toilet ?toilet)
    prec: (poisoned ?actor) (at ?actor ?place) (has ?place ?toilet)
          (vomit-untried ?actor)
    add: (healthy ?actor)
    del: (poisoned ?actor) (vomit-untried ?actor) (sick ?actor)
         (dying ?actor)

op: (wake-up ?victim)
    con: (person ?victim)
    prec: (unconscious ?victim) (healthy ?victim) (wakeable ?victim)
    add:
    del: (unconscious ?victim) (wakeable ?victim)

op: (watch ?actor ?target)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
    prec:
    add:
    del:
```

A.1.2 Rear Window

```
op: (answer-door ?actor ?visitor ?apartment)
    con: (person ?actor) (person ?visitor) (place ?apartment)
         (neq ?actor ?visitor)
    prec: (at ?actor ?apartment) (knocked ?visitor ?apartment)
          (lives-in ?actor ?apartment)
    add: (at ?visitor ?apartment)
    del: (knocked ?visitor ?apartment)

op: (call ?caller ?target ?phone)
    con: (person ?caller) (person ?target) (neq ?caller ?target)
         (telephone ?phone)
```

```
    prec: (has ?caller ?phone)
    add:
    del:

op: (catch ?actor ?intruder ?apartment)
    con: (person ?actor) (person ?intruder) (place ?apartment)
         (neq ?actor ?intruder)
    prec: (lives-in ?actor ?apartment) (at ?actor ?apartment)
          (at ?intruder ?apartment) (free ?intruder)
    add: (caught ?actor ?intruder) (caught ?intruder)
    del: (free ?intruder)

op: (release ?actor ?intruder ?apartment ?guest)
    con: (person ?actor) (person ?intruder) (person ?guest)
         (place ?apartment) (neq ?actor ?intruder) (neq ?actor ?guest)
         (neq ?intruder ?guest)
    prec: (lives-in ?actor ?apartment) (at ?actor ?apartment)
          (at ?intruder ?apartment) (at ?guest ?apartment)
          (caught ?actor ?intruder)
    add: (free ?intruder)
    del: (caught ?actor ?intruder) (caught ?intruder)

op: (climb ?actor ?apartment ?from)
    con: (person ?actor) (place ?apartment) (place ?from)
         (neq ?apartment ?from)
    prec: (mobile ?actor) (at ?actor ?from) (free ?actor)
          (climbable ?apartment ?from) (nonresident ?actor ?from)
    add: (at ?actor ?apartment)
    del: (at ?actor ?from)

op: (die ?victim)
    con: (person ?victim)
    prec: (hurt ?victim) (unconscious ?victim)
    add: (dead ?victim)
    del: (alive ?victim)

op: (drive ?actor ?car)
    con: (person ?actor) (car ?car)
    prec:
    add:
    del:

op: (faint ?actor)
    con: (person ?actor)
    prec:
```

```
    add: (unconscious ?actor)
    del:

op: (fall ?actor)
    con: (person ?actor)
    prec: (off-balance ?actor)
    add: (hurt ?actor)
    del:

op: (fight ?actor ?intruder ?apartment)
    con: (person ?actor) (person ?intruder) (place ?apartment)
         (neq ?actor ?intruder)
    prec: (lives-in ?actor ?apartment) (at ?actor ?apartment)
          (at ?intruder ?apartment)
    add: (hurt ?intruder)
    del:

op: (find-compartment ?actor)
    con: (person ?actor)
    prec:
    add:
    del:

op: (find-clue ?actor)
    con: (person ?actor)
    prec:
    add:
    del:

op: (find-evidence ?actor ?evidence ?place)
    con: (person ?actor) (evidence ?evidence) (place ?place)
    prec: (at ?actor ?place) (at ?evidence ?place) (free ?actor)
    add: (has ?actor ?evidence)
    del: (at ?evidence ?place)

op: (force-open ?actor ?apartment)
    con: (person ?actor) (place ?apartment)
    prec: (locked ?apartment)
    add: (off-balance ?actor)
    del: (locked ?apartment)

op: (give ?actor ?item ?target)
    con: (person ?actor) (person ?target) (object ?item)
         (neq ?actor ?target)
    prec: (has ?actor ?item)
```

```
    add: (has ?target ?item)
    del: (has ?actor ?item)

op: (go ?actor ?from ?to)
    con: (person ?actor) (place ?from) (place ?to) (neq ?from ?to)
    prec: (at ?actor ?from) (mobile ?actor) (free ?actor)
          (nonresident ?actor ?from)
    add: (at ?actor ?to)
    del: (at ?actor ?from)

op: (grab ?actor ?item ?from)
    con: (person ?actor) (object ?item) (place ?from)
    prec: (on ?item ?from)
    add: (has ?actor ?item)
    del: (on ?item ?from)

op: (hide ?actor ?hidefrom ?place)
    con: (person ?actor) (person ?hidefrom) (neq ?actor ?hidefrom)
         (place ?place)
    prec: (at ?actor ?place) (at ?hidefrom ?place) (free ?actor)
    add: (hidden ?actor ?place)
    del:

op: (ignore-knock ?actor ?visitor ?apartment)
    con: (person ?actor) (person ?visitor) (place ?apartment)
         (neq ?actor ?visitor)
    prec: (knocked ?visitor ?apartment) (lives-in ?actor ?apartment)
    add:
    del:

op: (ignore-signal ?signaled ?signaler)
    con: (person ?signaled) (person ?signaler)
         (neq ?signaled ?signaler)
    prec: (sent-signal ?signaler ?signaled)
    add:
    del: (sent-signal ?signaler ?signaled)

op: (knock ?visitor ?apartment ?owner)
    con: (person ?visitor) (place ?apartment) (person ?owner)
         (neq ?visitor ?owner)
    prec: (lives-in ?owner ?apartment) (at ?owner ?apartment)
    add: (knocked ?visitor ?apartment)
    del:

op: (knock-out ?actor ?victim ?apartment)
```

```
    con: (person ?actor) (person ?victim) (place ?apartment)
         (neq ?actor ?victim)
    prec: (at ?actor ?apartment) (at ?victim ?apartment)
          (hurt ?victim)
    add: (unconscious ?victim)
    del:

op: (lose-balance ?actor)
    con: (person ?actor)
    prec:
    add: (off-balance ?actor)
    del:

op: (open ?actor ?door)
    con: (person ?actor) (door ?door)
    prec:
    add: (open ?door)
    del:

op: (close ?actor ?door)
    con: (person ?actor) (door ?door)
    prec: (open ?door)
    add:
    del: (open ?door)

op: (park-car ?driver ?location ?car)
    con: (person ?driver) (place ?location) (car ?car)
    prec: (in ?driver ?car)
    add: (at ?driver ?location)
    del: (in ?driver ?car)

op: (pass-out ?actor)
    con: (person ?actor)
    prec: (hurt ?actor)
    add: (unconscious ?actor)
    del:

op: (receive-signal ?signaled ?signaler ?sendplace ?receiveplace)
    con: (person ?signaled) (person ?signaler)
         (neq ?signaled ?signaler) (place ?sendplace)
         (place ?receiveplace) (neq ?sendplace ?receiveplace)
    prec: (sent-signal ?signaler ?signaled) (free ?signaled)
          (at ?signaler ?sendplace) (at ?signaled ?receiveplace)
    add: (received-signal ?signaled ?signaler)
    del: (sent-signal ?signaler ?signaled)
```

```
op: (scream ?actor)
    con: (person ?actor)
    prec:
    add:
    del:

op: (search ?actor ?location)
    con: (person ?actor) (place ?location)
    prec: (at ?actor ?location)
    add:
    del:

op: (send-signal ?signaler ?signaled ?sendplace ?receiveplace)
    con: (person ?signaler) (person ?signaled)
         (neq ?signaled ?signaler) (place ?sendplace)
         (place ?receiveplace) (neq ?sendplace ?receiveplace)
    prec: (at ?signaler ?sendplace) (at ?signaled ?receiveplace)
          (signal-arranged ?signaler ?signaled)
    add: (sent-signal ?signaler ?signaled)
    del:

op: (talk ?speakerA ?speakerB ?location)
    con: (person ?speakerA) (person ?speakerB) (place ?location)
         (neq ?speakerA ?speakerB)
    prec: (at ?speakerA ?location) (at ?speakerB ?location)
    add: (talking ?speakerA ?speakerB) (talking ?speakerB ?speakerA)
    del:

op: (turn-off ?actor ?lights)
    con: (person ?actor) (light ?lights)
    prec: (on ?lights)
    add:
    del: (on ?lights)

op: (turn-on ?actor ?lights)
    con: (person ?actor) (light ?lights)
    prec:
    add: (on ?lights)
    del:

op: (unlock ?actor ?apartment)
    con: (person ?actor) (place ?apartment)
    prec: (locked ?apartment)
    add:
```

```
    del: (locked ?apartment)

op: (watch ?actor ?target)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
    prec:
    add:
    del:

op: (watch-together ?actor ?viewer ?target)
    con: (person ?actor) (person ?target) (person ?viewer)
         (neq ?actor ?target) (neq ?actor ?viewer)
         (neq ?viewer ?target)
    prec:
    add:
    del:

op: (whistle ?teakettle)
    con: (object ?teakettle)
    prec: (kettle ?teakettle)
    add:
    del:
```

A.1.3 Harry Potter and the Half-Blood Prince

```
op: (apparate ?actor ?from ?to)
    con: (person ?actor) (place ?from) (place ?to) (neq ?from ?to)
    prec: (headmaster ?actor) (at ?actor ?from) (conscious ?actor)
    add: (at ?actor ?to)
    del: (at ?actor ?from)

op: (apparate-with ?actor ?friend ?from ?to)
    con: (person ?actor) (person ?friend) (place ?from) (place ?to)
         (neq ?actor ?friend) (neq ?from ?to)
    prec: (headmaster ?actor) (at ?actor ?from) (at ?friend ?from)
          (conscious ?actor) (conscious ?friend)
    add: (at ?actor ?to) (at ?friend ?to)
    del: (at ?actor ?from) (at ?friend ?from)

op: (disarm ?actor ?target ?actorwand ?targetwand)
    con: (person ?actor) (person ?target) (wand ?actorwand)
         (wand ?targetwand) (neq ?actor ?target)
         (neq ?actorwand ?targetwand)
    prec: (has ?actor ?actorwand) (has ?target ?targetwand)
```

```
          (raised ?actorwand) (raised ?targetwand)
          (conscious ?actor) (conscious ?target)
    add: (disarmed ?targetwand)
    del: (has ?target ?targetwand) (raised ?targetwand)

op: (kill ?actor ?target ?actorwand)
    con: (person ?actor) (person ?target) (wand ?actorwand)
         (neq ?actor ?target)
    prec: (has ?actor ?actorwand) (conscious ?actor) (alive ?target)
          (raised ?actorwand)
    add: (killed-by ?target ?actor)
    del: (alive ?target) (conscious ?target)

op: (stupefy ?actor ?target ?actorwand)
    con: (person ?actor) (person ?target) (wand ?actorwand)
         (neq ?actor ?target)
    prec: (has ?actor ?actorwand) (conscious ?actor)
          (conscious ?target) (raised ?actorwand)
    add:
    del: (conscious ?target)

op: (crucio ?actor ?target ?actorwand)
    con: (person ?actor) (person ?target) (wand ?actorwand)
         (neq ?actor ?target)
    prec: (has ?actor ?actorwand) (conscious ?actor)
          (conscious ?target) (raised ?actorwand)
    add:
    del: (conscious ?target)

op: (raise-wand ?actor ?wand)
    con: (person ?actor) (wand ?wand)
    prec: (has ?actor ?wand) (conscious ?actor)
    add: (raised ?wand)
    del:

op: (lower-wand ?actor ?wand)
    con: (person ?actor) (wand ?wand)
    prec: (has ?actor ?wand) (raised ?wand) (conscious ?actor)
          (safe ?actor)
    add:
    del: (raised ?wand)

op: (point-wand ?actor ?target ?wand)
    con: (person ?actor) (wand ?wand) (person ?target)
         (neq ?actor ?target)
```

```
    prec: (has ?actor ?wand) (raised ?wand) (conscious ?actor)
    add: (threatened ?actor ?target)
    del:

op: (lower-wand-from ?actor ?wand ?target)
    con: (person ?actor) (wand ?wand) (person ?target)
         (neq ?actor ?target)
    prec: (has ?actor ?wand) (raised ?wand) (conscious ?actor)
          (threatened ?actor ?target) (safe ?actor)
    add:
    del: (raised ?wand) (threatened ?actor ?target)

op: (talk ?speakerA ?speakerB ?location)
    con: (person ?speakerA) (person ?speakerB) (place ?location)
         (neq ?speakerA ?speakerB)
    prec: (at ?speakerA ?location) (at ?speakerB ?location)
    add: (talking ?speakerA ?speakerB) (talking ?speakerB ?speakerA)
    del:

op: (talk-down ?actor ?target)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
    prec: (threatened ?actor ?target)
    add:
    del:

op: (plead ?actor ?target)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
    prec:
    add:
    del:

op: (go ?actor ?from ?to)
    con: (person ?actor) (hogwarts-place ?from) (hogwarts-place ?to)
         (neq ?from ?to)
    prec: (at ?actor ?from)
    add: (at ?actor ?to)
    del: (at ?actor ?from)

op: (use-passage ?actor ?from ?to)
    con: (person ?actor) (passage-place ?from) (passage-place ?to)
         (neq ?from ?to)
    prec: (at ?actor ?from)
    add: (at ?actor ?to)
    del: (at ?actor ?from)
```

```
op: (send-for-help ?actor ?sent ?helper ?place)
    con: (person ?actor) (person ?sent) (person ?helper)
         (place ?place) (neq ?actor ?sent)
         (neq ?actor ?helper) (neq ?sent ?helper)
    prec: (at ?actor ?place) (at ?sent ?place)
    add:
    del:

op: (hide ?actor)
    con: (person ?actor)
    prec:
    add: (hidden ?actor)
    del:

op: (hear ?actor ?target)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
    prec:
    add:
    del:

op: (watch ?actor ?target)
    con: (person ?actor) (person ?target) (neq ?actor ?target)
    prec: (hidden ?actor)
    add:
    del:

op: (pick-up ?actor ?wand)
    con: (person ?actor) (wand ?wand)
    prec: (disarmed ?targetwand)
    add: (has ?actor ?wand)
    del: (disarmed ?targetwand)
```

A.2 Scripts and Plans

Figure 25 and Figure 26 show the scripts used during evaluation for *Casino Royale* and *Rear Window*, respectively. Each was created using the knowledge acquisition and engineering process described in Chapter 4. Figure 27 shows the character plan used during evaluation for *Harry Potter and the Half-Blood Prince*, shown here as a script.

The black arrows represent temporal links, while the red dashed arrows show causal links. Exit nodes have red outlines. These diagrams does not show the co-designation constraints.

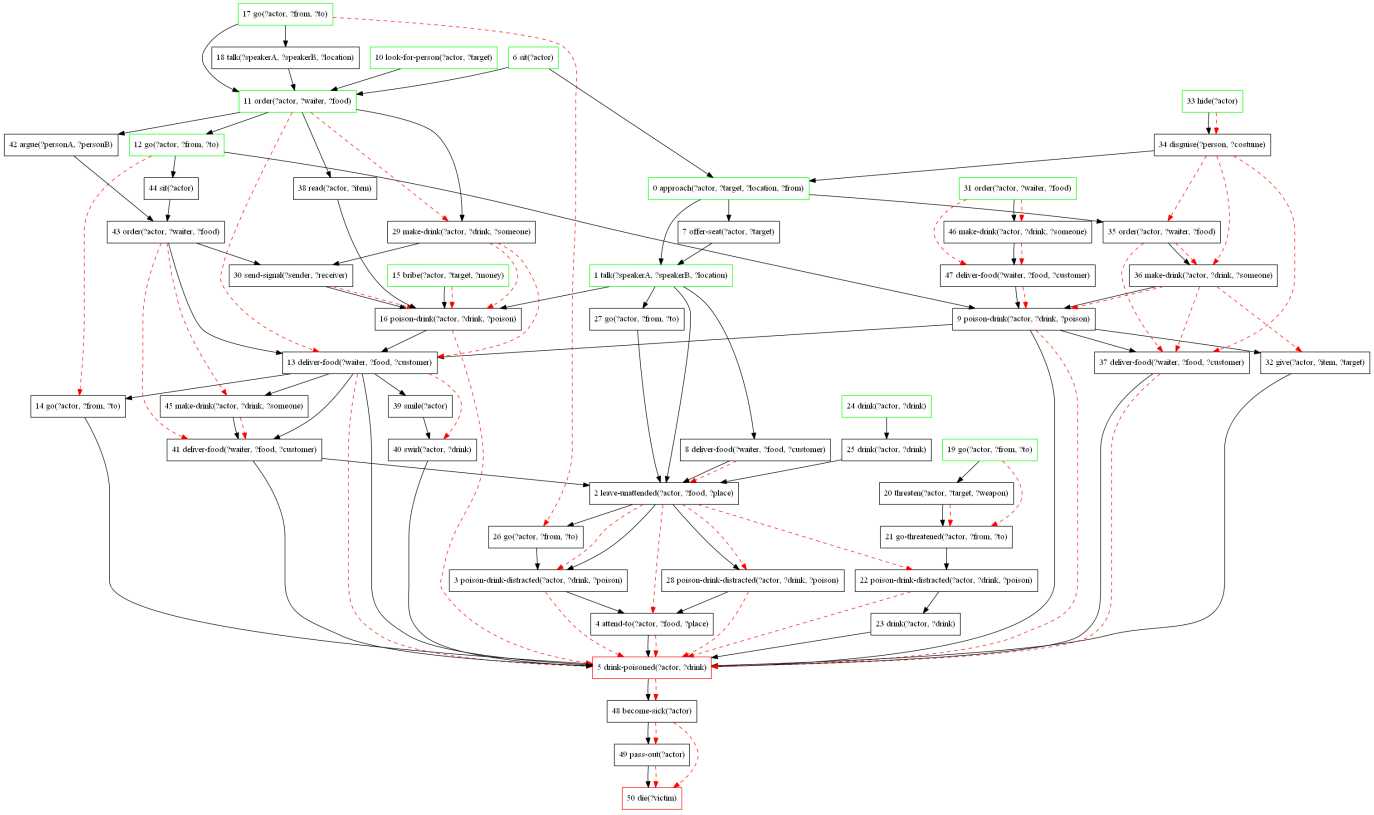

Figure 25: Script for *Casino Royale*

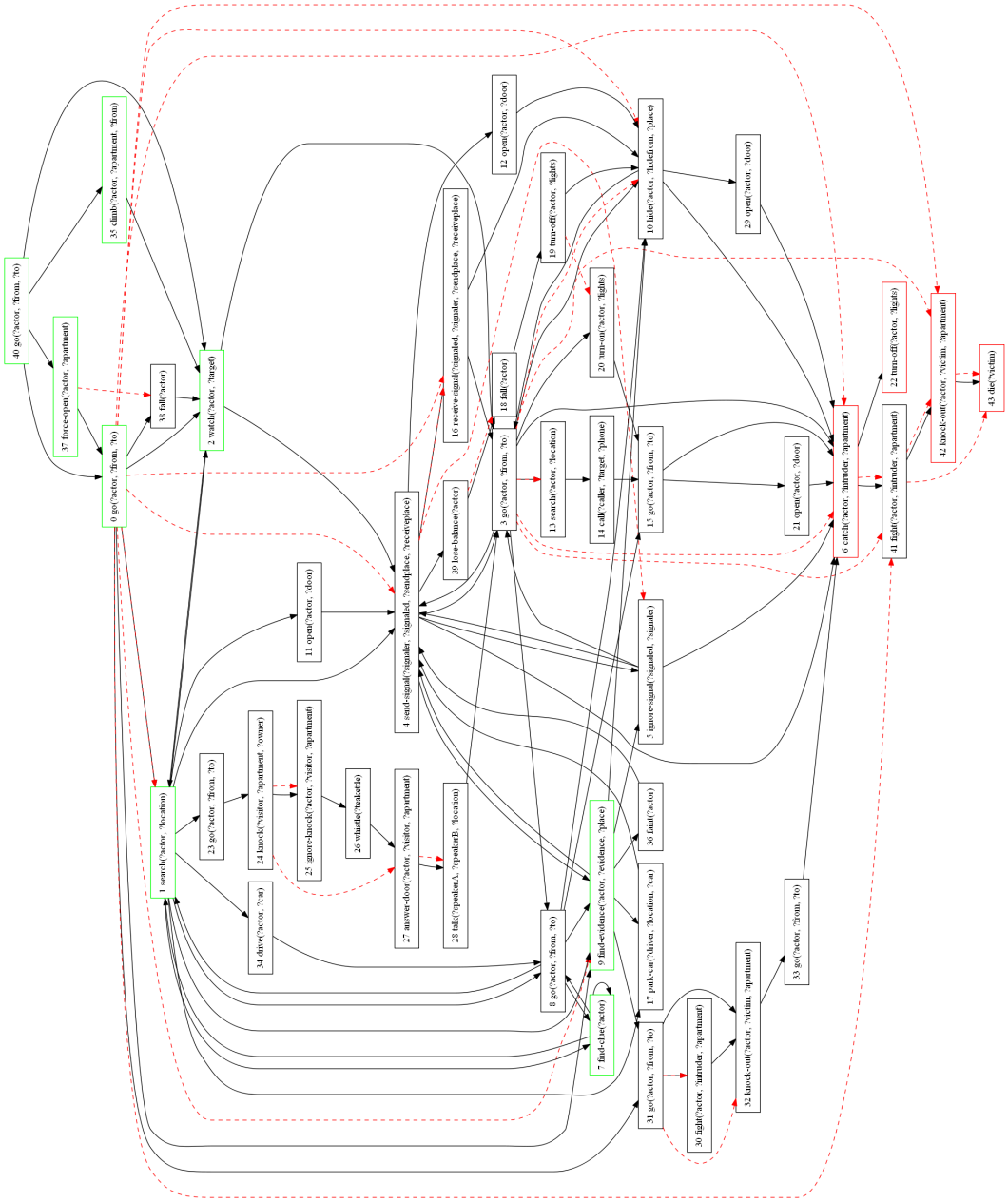

Figure 26: Script for *Rear Window*

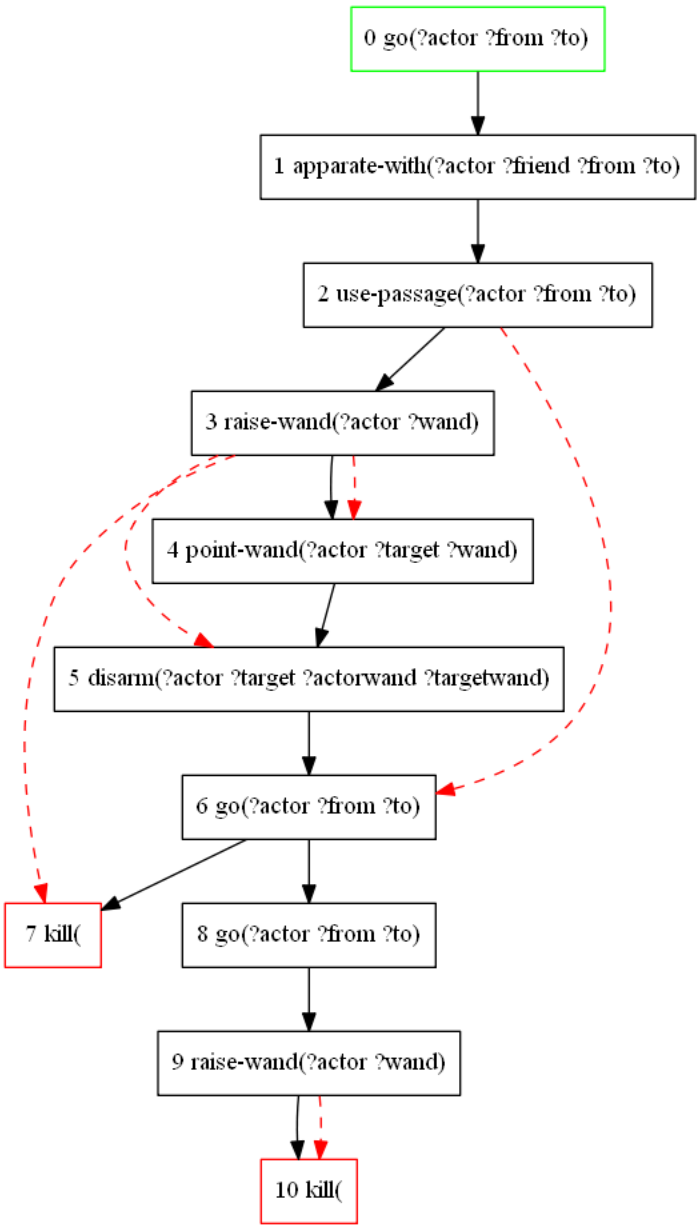

Figure 27: Character plan for *Harry Potter and the Half-Blood Prince*

APPENDIX B

MATERIALS

B.1 Knowledge Acquisition

Table 7: Knowledge Acquisition Study prompts

Spy 1	Start: A spy is at a bar or restaurant. Finish: The spy drinks from a drink poisoned by the villain.
Spy 2	Start: A spy is at a bar or restaurant. The spy just drank from a drink that was poisoned by the villain. Finish: The spy is no longer poisoned.
Rear Window	Start: A man (Tom) and a woman (Erin) suspect their neighbor of committing murder. Tom cannot leave the apartment, but Erin has just left the apartment to sneak into their neighbor's apartment to find proof. Tom and Erin have an agreed upon signal for if the neighbor is on his way home. Finish: The neighbor catches Erin in his apartment.

B.2 Evaluation 1 Stories

This section contains the natural language stories that participants read in Evaluation 1.

B.2.1 Casino Royale Original Version

The tray of drinks sat unattended on the bar. Valenka used this opportunity to poison one of them.

"Excuse me, please." The waitress picked up the tray of drinks and carried it over to the poker table. Valenka turned to watch.

The players continued the hand as the waitress dropped the drink in front of Bond. He saw Vesper sitting at the bar as he took a sip. Le Chiffre watched Bond from

across the poker table.

As Le Chiffre stared at him, Bond began to feel ill. He looked down at the drink, realizing he had been poisoned, and back at Le Chiffre. Bond quickly excused himself from the poker table. As he walked away, he grabbed a glass and a salt shaker from a nearby table. Vesper, Bond's partner, wondered why he was leaving the poker table now.

Bond staggered into the bathroom. He made his way to the sink, poured a glass of water, and emptied the salt shaker into the water. He chugged the mixture, and turned and vomited into the sink. It did not work.

He staggered out of the casino into the busy street. Cars and motorcycles nearly clipped him until he reached his car. He opened the glove compartment and retrieved the sensor that he was given before the mission. He activated it by injecting a needle into his arm.

In London, several alarms informed MI6 that James Bond was poisoned. One of the agents was quickly on the line with Bond giving him careful instructions. "Stay calm, and don't interrupt, because you'll be dead within two minutes if you don't do what I tell you. Remove the defibrillator from the pouch."

The agent turned to ask a coworker if they had identified the poison. They had not. Meanwhile, Bond did as he was told. The agent continued, "Now attach the leads to your chest." Bond fought through the pain and stuck the defibrillator leads onto his chest. The defibrillator began to charge. Bond's thumb sat ready over the red button that would activate the defibrillator.

"As soon as it reads charged—"

"Wait!" One of the agents in London cut off the instructions. "Bond, don't push the red button yet!"

The two agents began to argue with each other quietly. "His heart is going to stop."

"He only has time for one charge before he passes out."

The second agent went back to giving Bond instructions. "Take the blue epi-pen. Mid-neck, into the vein. That will counteract the poison." Bond jabbed the pen into his neck.

"You're going to pass out in a few seconds, and you need to keep your heart going. Push the red button now, Bond." Bond pushed the button and nothing happened. He pressed it again and again, and still did not feel anything. "Push the button NOW," the agent urged. Bond continued pressing to no avail until he noticed the loose wire coming out of the defibrillator. He picked it up and tried to reattach it to the lead on his chest. Before he could finish, Bond passed out. The agents in London could see his vital signs dropping.

Vesper came out to the car and saw Bond slumped over in the seat. She grabbed the loose wire and reinserted it into the lead on Bond's chest. She charged the defibrillator and activated it as soon as she could. Vesper pushed the red button, and Bond woke up.

B.2.2 Casino Royale Alternate Version

The tray of drinks sat unattended on the bar. Valenka used this opportunity to poison one of them.

"Excuse me, please." The waitress picked up the tray of drinks and carried it over to the poker table. Valenka turned to watch.

The players continued the hand as the waitress dropped the drink in front of Bond. He saw Vesper sitting at the bar as he took a sip. Le Chiffre watched Bond from across the poker table.

As Le Chiffre stared at him, Bond began to feel ill. He looked down at the drink, realizing he had been poisoned, and back at Le Chiffre. Bond quickly excused himself from the poker table to get to the medical gear in his car. Vesper, Bond's partner,

wondered why he was leaving the poker table now.

He staggered out of the casino into the busy street. Cars and motorcycles nearly clipped him until he reached his car. He opened the glove compartment and retrieved the sensor that he was given before the mission. He activated it by injecting a needle into his arm.

In London, several alarms informed MI6 that James Bond was poisoned. One of the agents was quickly on the line with Bond giving him careful instructions. "Stay calm, and don't interrupt, because you'll be dead within two minutes if you don't do what I tell you. Remove the defibrillator from the pouch."

The agent turned to ask a coworker if they had identified the poison. They had not. Meanwhile, Bond did as he was told. The agent continued, "Now attach the leads to your chest." Bond fought through the pain and stuck the defibrillator leads onto his chest. The defibrillator began to charge. Bond's thumb sat ready over the red button that would activate the defibrillator.

"Before you use the defibrillator, take the blue epi-pen. Mid-neck, into the vein. That will counteract the poison." Bond jabbed the pen into his neck.

"You're going to pass out in a few seconds, and you need to keep your heart going. Push the red button now, Bond." Bond pushed the red button and felt the jolt of electricity surge through him. The agents in London watched his vital signs return to normal.

B.2.3 Rear Window Original Version

Thorwald left his apartment, locking the door behind him. Jeff and Lisa knew this was their chance. If they were ever going to find proof that Thorwald had murdered his wife, Lisa would have to sneak into Thorwald's apartment right now.

Jeff watched from his wheelchair as Lisa ran through the courtyard. She climbed up the fire escape on to Thorwald's second floor balcony. Lisa then stepped carefully

from the balcony into Thorwald's open window. She ran through the apartment into the bedroom. Jeff held his camera up to his eyes so that he could get a better view of Lisa through his telephoto lens.

Lisa opened the suitcase on the bed and pulled out the handbag that they had seen earlier, holding it up so that Jeff could see it clearly from his own apartment. She opened it and pawed through it, but realized the bag was empty. She showed Jeff by opening the bag wide, holding it upside-down, and shaking it. Speaking to himself, Jeff urged Lisa to get out of the apartment. Lisa began searching through the dresser drawers as Stella came back into Jeff's apartment. "She said to call Thorwald's phone the second you see him come back," Stella told Jeff.

"I'm going to call it right now."

"No, no, give it another minute." Jeff put the phone down. Together, they watched Lisa continue to search the dresser.

Lisa came back into the living room and held up the necklace she had found in the dresser so Jeff and Stella could see it clearly. At the same time, they saw Thorwald walking down the hallway to his apartment. It was too late for them to call his phone. Lisa headed for the door to leave. As she approached the door, she heard Thorwald coming. She ran back to the bedroom to find someplace to hide. Jeff and Stella lost sight of her just as they saw Thorwald enter his apartment. Jeff called the police.

"Hello. Look, a man is assaulting a woman at 125 West Ninth Street. Second floor, the rear of the building. Make it fast."

As Jeff spoke to the police, Thorwald walked through his apartment. In his room, he noticed that the suitcase had been rifled through, and that his wife's handbag had been removed. He heard a noise behind him, and turned around. As Thorwald advanced toward the noise, Jeff could see Lisa backing her way through the apartment. She held something behind her back and gestured while saying something to Thorwald. Jeff and Stella couldn't make out what she was saying. As she turned to

leave the apartment, Thorwald grabbed her wrist and threw her to the ground. Jeff and Stella watched, unable to do a thing.

Thorwald held out his hand toward Lisa, urging her to give him what she had. Lisa handed him the necklace, which he put in his coat pocket. Lisa stood up again and tried to leave, but Thorwald grabbed her again. "I told you!" she shouted. She struggled to get free, shouting "Let go of me!" As she struggled, she turned and looked out the window and shouted Jeff's name. Thorwald kept his grasp on her as he reached with one arm to shut off the lights in the apartment.

"Stella, what do we do?"

Lisa continued to shout for Jeff. He and Stella could see Lisa and Thorwald wrestling in the darkness. Stella saw the police coming down the hallway.

The police knocked on the door. Thorwald released Lisa and answered the door.

B.2.4 Rear Window Alternate Version

Thorwald left his apartment, locking the door behind him. Jeff and Lisa knew this was their chance. If they were ever going to find proof that Thorwald had murdered his wife, Lisa would have to sneak into Thorwald's apartment right now.

Jeff watched from his wheelchair as Lisa ran through the courtyard. She climbed up the fire escape on to Thorwald's second floor balcony. Lisa then stepped carefully from the balcony into Thorwald's open window. She ran through the apartment into the bedroom. Jeff held his camera up to his eyes so that he could get a better view of Lisa through his telephoto lens.

Lisa opened the suitcase on the bed and pulled out the handbag that they had seen earlier, holding it up so that Jeff could see it clearly from his own apartment. She opened it and pawed through it, but realized the bag was empty. She showed Jeff by opening the bag wide, holding it upside-down, and shaking it. Speaking to himself, Jeff urged Lisa to get out of the apartment. Lisa began searching through

the dresser drawers.

Jeff decided to call the police. He made up a story to get them to come. “Hello. Look, a man is assaulting a woman at 125 West Ninth Street. Second floor, the rear of the building. Make it fast.”

As Jeff spoke to the police, he watched Lisa continue to search the dresser. Lisa came back into the living room and held up the necklace she had found in the dresser so Jeff could see it clearly. At the same time, he saw Thorwald walking down the hallway to his apartment. Lisa headed for the door to leave. As she approached the door, she heard Thorwald coming. She ran back to the bedroom to find someplace to hide. Jeff lost sight of her just as he saw Thorwald enter his apartment.

Thorwald walked through his apartment. In his room, he noticed that the suitcase had been rifled through, and that his wife’s handbag had been removed. He heard a noise behind him, and turned around. As Thorwald advanced toward the noise, Jeff could see Lisa backing her way through the apartment. She held something behind her back and gestured while saying something to Thorwald. Jeff couldn’t make out what she was saying. As she turned to leave the apartment, Thorwald grabbed her wrist and threw her to the ground. Jeff watched, unable to do a thing.

Thorwald held out his hand toward Lisa, urging her to give him what she had. Lisa handed him the necklace, which he put in his coat pocket. Lisa stood up again and tried to leave, but Thorwald grabbed her again. “I told you!” she shouted. She struggled to get free, shouting “Let go of me!” As she struggled, she turned and looked out the window and shouted Jeff’s name. Thorwald kept his grasp on her as he reached with one arm to shut off the lights in the apartment.

Lisa continued to shout for Jeff. He could see Lisa and Thorwald wrestling in the darkness. Then, Jeff saw the police coming down the hallway.

The police knocked on the door. Thorwald released Lisa and answered the door.

B.2.5 Harry Potter Original Version

Draco left his room in Slytherin. He knew Dumbledore would appear soon in the astronomy tower. He had told Bellatrix how to use the passage between Borgin & Burkes and the Room of Requirement. Soon, she would join him in the tower, right before he killed Dumbledore. And if he couldn't do it, Snape would take care of it for him.

Meanwhile, Harry and Dumbledore apparated into the astronomy tower. Harry wanted to take him to the hospital, but Dumbledore gave him other instructions. "Snape. Wake him. Tell him what happened. Speak to no one else."

Harry walked to the stairs, while Dumbledore began to sit. When Harry reached the stairs, he heard noises coming from below. Dumbledore told Harry to hide and stay quiet until he said so, no matter what happened. Harry hesitated, but listened to the professor.

Harry went below the top of the tower. He hid and saw Draco climbing the tower stairs.

"Good evening, Draco. What brings you here this fine spring evening?" Dumbledore said.

Draco held his wand up, pointed directly at Dumbledore. "Who else is here? I heard you talking."

"I often talk aloud to myself."

Neither spoke. Harry continued to watch from below. Draco stepped around the observatory with his wand still pointed at the professor. "Draco, you are no assassin."

"How do you know what I am? I've done things that would shock you!"

"Like cursing Katie Bell and hoping that in return she'd bare a cursed necklace to me? Like replacing a bottle of mead with one laced with poison? Forgive me, but I cannot help feeling that these actions are so weak that your heart can't really have been in them."

"He trusts me. I was chosen." Draco raised his sleeve to show the Dark Mark on his arm.

Dumbledore raised his arms and spoke to Draco. "I shall make it easy for you"

"*Expelliarmus*!" Draco disarmed Dumbledore of his wand, which flew several feet away.

"Very good. Very good." As Dumbledore spoke, the noise of the door opening and closing could be heard below. "There are others?" Dumbledore asked. Draco kept his wand pointed at the professor. "How?"

"The vanishing cabinet in the Room of Requirement. I've been mending it."

"Let me guess—it has a twin."

"In Borgin & Burkes. They form a passage."

"Ingenious! Draco, years ago, I knew a boy who made all the wrong choices. Please let me help you," Dumbledore pleaded.

"I don't want your help. Don't you understand? I have to do this. I have to kill you. Or he's going to kill me."

Harry moved to keep himself hidden as the others climbed the stairs. Draco held his wand toward Dumbledore when Bellatrix reached the top. "Well, look what we have here," she said. "Well done, Draco."

"Good evening, Bellatrix," Dumbledore said.

"Do it!" urged Bellatrix.

Harry stood below and raised his wand up. He had a clear shot at Draco. Suddenly, Snape appeared behind him, pointing his wand at Harry. Snape raised his finger to his lips, telling Harry to keep quiet. Harry and Snape both lowered their wands.

"Go on, Draco!" Bellatrix said. "NOW!"

Snape climbed the stairs to the top of the observatory, appearing behind Draco. "No," said Snape. Draco lowered his wand and stepped out of the way. Dumbledore

watched Harry below, but said nothing to him.

"Snape. Please," said Dumbledore.

Snape raised his wand and pointed it at Dumbledore. "*Avada Kedavra*!" Dumbledore was struck and killed immediately, and his body fell from the tower to the ground below as Harry watched. Draco, Snape, and the others fled from the tower.

B.2.6 Harry Potter Alternate Version

Draco left his room in Slytherin. He knew Dumbledore would appear soon in the astronomy tower. He had told Bellatrix how to use the passage between Borgin & Burkes and the Room of Requirement. Soon, she would join him in the tower, right before he killed Dumbledore. And if he couldn't do it, Snape would take care of it for him.

Meanwhile, Harry and Dumbledore apparated into the astronomy tower. Harry wanted to take him to the hospital, but Dumbledore gave him other instructions. "Snape. Wake him. Tell him what happened. Speak to no one else."

Harry walked to the stairs, while Dumbledore began to sit. When Harry reached the stairs, he heard noises coming from below. Dumbledore told Harry to hide and stay quiet until he said so, no matter what happened. Harry hesitated, but listened to the professor.

Harry went below the top of the tower. He hid and saw Draco climbing the tower stairs.

"Good evening, Draco. What brings you here this fine spring evening?" Dumbledore said.

Draco held his wand up, pointed directly at Dumbledore. "Who else is here? I heard you talking."

"I often talk aloud to myself."

Neither spoke. Harry continued to watch from below. Draco stepped around the

observatory with his wand still pointed at the professor. "Draco, you are no assassin."

"How do you know what I am? I've done things that would shock you!"

"Like cursing Katie Bell and hoping that in return she'd bare a cursed necklace to me? Like replacing a bottle of mead with one laced with poison? Forgive me, but I cannot help feeling that these actions are so weak that your heart can't really have been in them."

"He trusts me. I was chosen." Draco raised his sleeve to show the Dark Mark on his arm.

Dumbledore raised his arms and pointed his wand at Draco.

Dumbledore and Draco heard the noise of the door opening and closing could beneath them. "There are others?" Dumbledore asked. Draco kept his wand pointed at the professor. "How?"

"The vanishing cabinet in the Room of Requirement. I've been mending it."

"Let me guess—it has a twin."

"In Borgin & Burkes. They form a passage."

"Ingenious! Draco, years ago, I knew a boy who made all the wrong choices. Please let me help you," Dumbledore pleaded.

"I don't want your help. Don't you understand? I have to do this. I have to kill you. Or he's going to kill me."

Harry moved to keep himself hidden as the others climbed the stairs. Draco held his wand toward Dumbledore when Bellatrix reached the top. "Well, look what we have here," she said. "Well done, Draco."

"Good evening, Bellatrix," Dumbledore said.

"Do it!" urged Bellatrix.

Harry stood below and raised his wand up. He had a clear shot at Draco. Suddenly, Snape appeared behind him, pointing his wand at Harry. Snape raised his finger to his lips, telling Harry to keep quiet. Harry and Snape both lowered their

wands.

"Go on, Draco!" Bellatrix said. "NOW!"

Snape climbed the stairs to the top of the observatory, appearing behind Draco. "No," said Snape. Draco lowered his wand and stepped out of the way. Dumbledore watched Harry below, but said nothing to him.

Dumbledore turned to Snape and aimed his wand. Snape raised his wand and pointed it at Dumbledore. "*Avada Kedavra*!" Dumbledore was struck and killed immediately, and his body fell from the tower to the ground below as Harry watched. Draco, Snape, and the others fled from the tower.

B.3 Evaluation 1 Questions

The following three forms were presented to participants during Evaluation 1. Table 8 shows the form used to indicate which story out of a pair was more suspenseful. Table 9 shows the form used to indicate the Likert rating for how suspenseful a story was.

Additionally, Table 10 shows the form used to respond to story comprehension questions, which was only given to sixteen participants. In this form, "Story C" and "Story D" refer to the *Rear Window* stories, while "Story E" and "Story F" refer to the *Harry Potter* stories.

B.4 Evaluation Time-Slices

The following sets of time-slices were provided to Dramatis in all three evaluations.

B.4.1 Casino Royale Original Version

```
Action: (poison-drink-distracted Valenka vodkaMartini inkPoison)
Location: Casino Royale
Characters: Waitress, Valenka
Dialogue: none
Effects: (unattended vodkaMartini) (at vodkaMartini bar)
         (belongs-to vodkaMartini Waitress) (edible vodkaMartini)
         (drink vodkaMartini) (place bar) (person Waitress)
```

Table 8: Participant Form for Story Preference

For each pair of stories, please indicate, by circling, which story was **more suspenseful.**

Pair 1

Story A Story B

Pair 2

Story C Story D

Pair 3

Story E Story F

Table 9: Participant Form for Likert rating

For the story you just read, please indicate **how suspenseful** the story was by circling a number below.

(least suspenseful) 1 ---- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7 (most suspenseful)

Table 10: Participant Form for Story Comprehension Questions

What difference, if any, was there between **Story C** and **Story D**?

What difference, if any, was there between **Story E** and **Story F**?

```
          (at Waitress bar) (evil Valenka) (poisonous inkPoison)
          (waiter Waitress) (ordered vodkaMartini James_Bond)
          (person James_Bond) (vomit-untried James_Bond)
          (conscious Waitress) (conscious James_Bond)
          (conscious Valenka) (place bathroom) (has bathroom sink)
          (toilet sink)

Action: (attend-to Waitress vodkaMartini bar)
Location: Casino Royale
Characters: Waitress, Valenka
Dialogue: none
Effects: none

Action: (watch Valenka Waitress)
Location: Casino Royale
Characters: Waitress, Valenka
Dialogue: none
Effects: none

Action: (deliver-food Waitress vodkaMartini James_Bond)
Location: Casino Royale
Characters: Waitress, James_Bond
Dialogue: none
Effects: (at James_Bond pokerTable) (place pokerTable)

Action: (drink-poisoned James_Bond vodkaMartini)
Location: Casino Royale
Characters: James_Bond
Dialogue: none
Effects: none

Action: (become-sick James_Bond)
Location: Casino Royale
Characters: James_Bond
Dialogue: none
Effects: none

Action: (grab James_Bond salt otherTable)
Location: Casino Royale
Characters: James_Bond
Dialogue: none
Effects: (object salt) (place otherTable)

Action: (watch Vesper_Lynd James_Bond)
Location: Casino Royale
```

```
Characters: Vesper_Lynd, James_Bond
Dialogue: none
Effects: (person Vesper_Lynd) (at Vesper_Lynd bar)
         (conscious Vesper_Lynd)

Action: (go James_Bond pokerTable bathroom)
Location: Casino Royale
Characters: James_Bond
Dialogue: none
Effects:

Action: (vomit James_Bond bathroom sink)
Location: Casino Royale
Characters: James_Bond
Dialogue: none
Effects: (poisoned James_Bond) (sick James_Bond)
         (dying James_Bond) !(healthy James_Bond)

Action: (go James_Bond bathroom AstonMartin)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: (place AstonMartin)

Action: (contact James_Bond MI6)
Location: Outside
Characters: James_Bond, MI6
Dialogue: none
Effects: (person MI6)

Action: (talk-b MI6 James_Bond)
Location: Outside
Characters: James_Bond, MI6
Dialogue: (dying James_Bond) (at defib AstonMartin)
Effects: (defibrillator defib)

Action: (attach-leads James_Bond James_Bond defib AstonMartin)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: none

Action: (talk-b MI6 James_Bond)
Location: Outside
Characters: James_Bond, MI6
```

```
Dialogue: (object BlueEpiPen) (at BlueEpiPen AstonMartin)
          (antidote BlueEpiPen)
Effects: none

Action: (grab James_Bond BlueEpiPen AstonMartin)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: none

Action: (apply-antidote James_Bond BlueEpiPen James_Bond)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: none

Action: (talk-b MI6 James_Bond)
Location: Outside
Characters: James_Bond, MI6
Dialogue: none
Effects: none

Action: (defibrillate James_Bond defib James_Bond)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: (dying James_Bond) !(wakeable James_Bond)

Action: (defibrillate James_Bond defib James_Bond)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: (dying James_Bond) !(wakeable James_Bond)
         !(leads-attached defib James_Bond)

Action: (attach-leads James_Bond James_Bond defib AstonMartin)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: !(leads-attached defib James_Bond)

Action: (pass-out James_Bond)
Location: Outside
Characters: James_Bond
Dialogue: none
```

```
Effects: none

Action: (go Vesper_Lynd bar AstonMartin)
Location: Outside
Characters: Vesper_Lynd
Dialogue: none
Effects: none

Action: (attach-leads Vesper_Lynd James_Bond defib AstonMartin)
Location: Outside
Characters: Vesper_Lynd, James_Bond
Dialogue: none
Effects: none

Action: (defibrillate Vesper_Lynd defib James_Bond)
Location: Outside
Characters: Vesper_Lynd, James_Bond
Dialogue: none
Effects: none
```

B.4.2 Casino Royale Alternate Version

```
Action: (poison-drink-distracted Valenka vodkaMartini inkPoison)
Location: Casino Royale
Characters: Waitress, Valenka
Dialogue: none
Effects: (unattended vodkaMartini) (at vodkaMartini bar)
         (belongs-to vodkaMartini Waitress) (edible vodkaMartini)
         (drink vodkaMartini) (place bar) (person Waitress)
         (at Waitress bar) (evil Valenka) (poisonous inkPoison)
         (waiter Waitress) (ordered vodkaMartini James_Bond)
         (person James_Bond) (vomit-untried James_Bond)
         (conscious Waitress) (conscious James_Bond)
         (conscious Valenka) (place bathroom) (has bathroom sink)
         (toilet sink)

Action: (attend-to Waitress vodkaMartini bar)
Location: Casino Royale
Characters: Waitress, Valenka
Dialogue: none
Effects: none

Action: (watch Valenka Waitress)
```

```
Location: Casino Royale
Characters: Waitress, Valenka
Dialogue: none
Effects: none

Action: (deliver-food Waitress vodkaMartini James_Bond)
Location: Casino Royale
Characters: Waitress, James_Bond
Dialogue: none
Effects: (at James_Bond pokerTable) (place pokerTable)

Action: (drink-poisoned James_Bond vodkaMartini)
Location: Casino Royale
Characters: James_Bond
Dialogue: none
Effects: none

Action: (become-sick James_Bond)
Location: Casino Royale
Characters: James_Bond
Dialogue: none
Effects: (place AstonMartin) (object BlueEpiPen)
         (at BlueEpiPen AstonMartin) (defibrillator defib)

Action: (watch Vesper_Lynd James_Bond)
Location: Casino Royale
Characters: Vesper_Lynd, James_Bond
Dialogue: none
Effects: (person Vesper_Lynd) (at Vesper_Lynd bar)
         (conscious Vesper_Lynd)

Action: (go James_Bond pokerTable AstonMartin)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: none

Action: (contact James_Bond MI6)
Location: Outside
Characters: James_Bond, MI6
Dialogue: none
Effects: (person MI6)

Action: (talk-b MI6 James_Bond)
Location: Outside
```

```
Characters: James_Bond, MI6
Dialogue: (dying James_Bond) (at defib AstonMartin)
Effects: (defibrillator defib)

Action: (attach-leads James_Bond James_Bond defib AstonMartin)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: none

Action: (talk-b MI6 James_Bond)
Location: Outside
Characters: James_Bond, MI6
Dialogue: (antidote BlueEpiPen)
Effects: none

Action: (grab James_Bond BlueEpiPen AstonMartin)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: none

Action: (apply-antidote James_Bond BlueEpiPen James_Bond)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: none

Action: (talk-b MI6 James_Bond)
Location: Outside
Characters: James_Bond, MI6
Dialogue: none
Effects: none

Action: (defibrillate James_Bond defib James_Bond)
Location: Outside
Characters: James_Bond
Dialogue: none
Effects: none
```

B.4.3 Rear Window Original Version

```
Action: (go Thorwald ThorApt out)
```

```
Location: Thor Apt
Characters: Thorwald
Dialogue: none
Effects: (place ThorApt) (lives-in Thorwald ThorApt) (person Thorwald)
         (mobile Thorwald) (free Thorwald) (nonresident Thorwald out)

Action: (watch Jeff Lisa)
Location: Jeff Apt
Characters: Jeff, Lisa
Dialogue: none
Effects: (person Jeff) (person Lisa) (place JeffApt) (place courtyard)
         (at Jeff JeffApt) (at Lisa courtyard) (lives-in Jeff JeffApt)
         (mobile Lisa) (free Jeff) (free Lisa)
         (nonresident Lisa ThorApt) (nonresident Lisa courtyard)
         (nonresident Lisa JeffApt) (nonresident Jeff courtyard)
         (nonresident Jeff ThorApt)

Action: (climb Lisa ThorApt courtyard)
Location: Courtyard
Characters: Lisa
Dialogue: none
Effects: (climbable ThorApt courtyard) (climbable courtyard ThorApt)

Action: (watch Jeff Lisa)
Location: Jeff Apt
Characters: Jeff, Lisa
Dialogue: none
Effects: none

Action: (search Lisa ThorApt)
Location: Thor Apt
Characters: Lisa
Dialogue: none
Effects: none

Action: (find-clue Lisa)
Location: Thor Apt
Characters: Lisa
Dialogue: none
Effects: none

Action: (search Lisa ThorApt)
Location: Thor Apt
Characters: Lisa
Dialogue: none
```

```
Effects: none

Action: (go Stella courtyard JeffApt)
Location: Jeff Apt
Characters: Jeff, Stella
Dialogue: (signal-arranged Jeff Lisa)
Effects: (on phone JeffApt) (object phone) (telephone phone)
         (person Stella) (free Stella) (mobile Stella)
         (nonresident Stella courtyard) (nonresident Stella JeffApt)
         (nonresident Stella ThorApt)

Action: (grab Jeff phone JeffApt)
Location: Jeff Apt
Characters: Jeff
Dialogue: none
Effects: none

Action: (find-evidence Lisa necklace ThorApt)
Location: Thor Apt
Characters: Lisa
Dialogue: none
Effects: (object necklace) (evidence necklace)

Action: (watch-together Jeff Stella Thorwald)
Location: Thor Apt
Characters: Jeff, Thorwald, Stella
Dialogue: none
Effects: (at Thorwald hallway) (place hallway)

Action: (go Lisa ThorApt hallway)
Location: Thor Apt
Characters: Lisa
Dialogue: none
Effects: !(at Lisa hallway) (at Lisa ThorApt)

Action: (hide Lisa Thorwald ThorApt)
Location: Thor Apt
Characters: Lisa, Thorwald
Dialogue: none
Effects: none

Action: (go Thorwald hallway ThorApt)
Location: Thor Apt
Characters: Thorwald
Dialogue: none
```

```
Effects: none

Action: (call Jeff Police phone)
Location: Jeff Apt
Characters: Jeff, Police
Dialogue: none
Effects: (mobile Police) (person Police) (free Police)
         (at Police policeStation) (place policeStation)
         (nonresident Police ThorApt)
         (nonresident Police policeStation)
         (nonresident Police courtyard) (nonresident Police JeffApt)

Action: (search Thorwald ThorApt)
Location: Thor Apt
Characters: Thorwald
Dialogue: none
Effects: none

Action: (catch Thorwald Lisa ThorApt)
Location: Thor Apt
Characters: Thorwald, Lisa
Dialogue: none
Effects: none

Action: (watch-together Jeff Stella Lisa)
Location: Jeff Apt
Characters: Jeff, Lisa, Stella
Dialogue: none
Effects: none

Action: (fight Thorwald Lisa ThorApt)
Location: Thor Apt
Characters: Lisa, Thorwald
Dialogue: none
Effects: none

Action: (watch-together Jeff Stella Lisa)
Location: Jeff Apt
Characters: Jeff, Lisa, Stella
Dialogue: none
Effects: none

Action: (give Lisa necklace Thorwald)
Location: Thor Apt
Characters: Lisa, Thorwald
```

```
Dialogue: none
Effects: none

Action: (fight Thorwald Lisa ThorApt)
Location: Thor Apt
Characters: Lisa, Thorwald
Dialogue: none
Effects: none

Action: (scream Lisa)
Location: Thor Apt
Characters: Lisa, Thorwald
Dialogue: none
Effects: none

Action: (turn-off Thorwald lights)
Location: Thor Apt
Characters: Thorwald, Lisa
Dialogue: none
Effects: (light lights)

Action: (talk Jeff Stella JeffApt)
Location: Jeff Apt
Characters: Jeff, Stella
Dialogue: none
Effects: none

Action: (scream Lisa)
Location: Thor Apt
Characters: Lisa
Dialogue: none
Effects: none

Action: (fight Thorwald Lisa ThorApt)
Location: Thor Apt
Characters: Thorwald, Lisa
Dialogue: none
Effects: none

Action: (watch Stella Police)
Location: Thor Apt
Characters: Stella, Police
Dialogue: none
Effects: (at Police hallway)
```

```
Action: (knock Police ThorApt Thorwald)
Location: Thor Apt
Characters: Thorwald, Police, Lisa
Dialogue: none
Effects: none

Action: (release Thorwald Lisa ThorApt Police)
Location: Thor Apt
Characters: Thorwald, Lisa
Dialogue: none
Effects: none
```

B.4.4 Rear Window Alternate Version

```
Action: (go Thorwald ThorApt out)
Location: Thor Apt
Characters: Thorwald
Dialogue: none
Effects: (place ThorApt) (lives-in Thorwald ThorApt) (person Thorwald)
         (mobile Thorwald) (free Thorwald) (nonresident Thorwald out)

Action: (watch Jeff Lisa)
Location: Jeff Apt
Characters: Jeff, Lisa
Dialogue: none
Effects: (person Jeff) (person Lisa) (place JeffApt) (place courtyard)
         (at Jeff JeffApt) (at Lisa courtyard) (lives-in Jeff JeffApt)
         (mobile Lisa) (free Jeff) (free Lisa)
         (nonresident Lisa ThorApt) (nonresident Lisa courtyard)
         (nonresident Lisa JeffApt) (nonresident Jeff courtyard)
         (nonresident Jeff ThorApt)

Action: (climb Lisa ThorApt courtyard)
Location: Courtyard
Characters: Lisa
Dialogue: none
Effects: (climbable ThorApt courtyard) (climbable courtyard ThorApt)

Action: (watch Jeff Lisa)
Location: Jeff Apt
Characters: Jeff, Lisa
Dialogue: none
Effects: none
```

```
Action: (search Lisa ThorApt)
Location: Thor Apt
Characters: Lisa
Dialogue: none
Effects: none

Action: (find-clue Lisa)
Location: Thor Apt
Characters: Lisa
Dialogue: none
Effects: none

Action: (search Lisa ThorApt)
Location: Thor Apt
Characters: Lisa
Dialogue: none
Effects: none

Action: (call Jeff Police phone)
Location: Jeff Apt
Characters: Jeff, Police
Dialogue: none
Effects: (on phone JeffApt) (object phone) (telephone phone)
         (mobile Police) (person Police) (free Police)
         (at Police policeStation) (place policeStation)
         (nonresident Police ThorApt)
         (nonresident Police policeStation)
         (nonresident Police courtyard) (nonresident Police JeffApt)

Action: (find-evidence Lisa necklace ThorApt)
Location: Thor Apt
Characters: Lisa
Dialogue: none
Effects: (object necklace) (evidence necklace)

Action: (watch Jeff Thorwald)
Location: Thor Apt
Characters: Jeff, Thorwald
Dialogue: none
Effects: (at Thorwald hallway) (place hallway)

Action: (go Lisa ThorApt hallway)
Location: Thor Apt
Characters: Lisa
```

```
Dialogue: none
Effects: !(at Lisa hallway) (at Lisa ThorApt)

Action: (hide Lisa Thorwald ThorApt)
Location: Thor Apt
Characters: Lisa, Thorwald
Dialogue: none
Effects: none

Action: (go Thorwald hallway ThorApt)
Location: Thor Apt
Characters: Thorwald
Dialogue: none
Effects: none

Action: (search Thorwald ThorApt)
Location: Thor Apt
Characters: Thorwald
Dialogue: none
Effects: none

Action: (catch Thorwald Lisa ThorApt)
Location: Thor Apt
Characters: Thorwald, Lisa
Dialogue: none
Effects: none

Action: (watch Jeff Lisa)
Location: Jeff Apt
Characters: Jeff, Lisa
Dialogue: none
Effects: none

Action: (fight Thorwald Lisa ThorApt)
Location: Thor Apt
Characters: Lisa, Thorwald
Dialogue: none
Effects: none

Action: (watch Jeff Lisa)
Location: Jeff Apt
Characters: Jeff, Lisa
Dialogue: none
Effects: none
```

```
Action: (give Lisa necklace Thorwald)
Location: Thor Apt
Characters: Lisa, Thorwald
Dialogue: none
Effects: none

Action: (fight Thorwald Lisa ThorApt)
Location: Thor Apt
Characters: Lisa, Thorwald
Dialogue: none
Effects: none

Action: (scream Lisa)
Location: Thor Apt
Characters: Lisa, Thorwald
Dialogue: none
Effects: none

Action: (turn-off Thorwald lights)
Location: Thor Apt
Characters: Thorwald, Lisa
Dialogue: none
Effects: (light lights)

Action: (scream Lisa)
Location: Thor Apt
Characters: Lisa
Dialogue: none
Effects: none

Action: (fight Thorwald Lisa ThorApt)
Location: Thor Apt
Characters: Thorwald, Lisa
Dialogue: none
Effects: none

Action: (watch Jeff Police)
Location: Thor Apt
Characters: Jeff, Police
Dialogue: none
Effects: (at Police hallway)

Action: (knock Police ThorApt Thorwald)
Location: Thor Apt
Characters: Thorwald, Police, Lisa
```

```
Dialogue: none
Effects: none

Action: (release Thorwald Lisa ThorApt Police)
Location: Thor Apt
Characters: Thorwald, Lisa
Dialogue: none
Effects: none
```

B.4.5 Harry Potter Original Version

```
Action: (go Draco slytherin tower)
Location: Slytherin
Characters: Draco
Dialogue: none
Plan: Draco:dracoplan
Effects: (person Draco) (place slytherin) (place tower)
         (hogwarts-place slytherin) (hogwarts-place tower)
         (conscious Draco) (has Draco DracoWand) !(at Draco tower)
         (wand DracoWand) (alive Draco) (at Bellatrix borgin)
         (passage-place borgin) (passage-place roomreqt)
         (place borgin) (place roomreqt) (hogwarts-place roomreqt)

Action: (apparate-with Dumbledore Harry hogsmeade tower)
Location: Tower
Characters: Harry, Dumbledore
Dialogue: none
Effects: (person Harry) (person Dumbledore) (place hogsmeade)
         (conscious Harry) (conscious Dumbledore)
         (headmaster Dumbledore) (has Harry HarryWand)
         (has Dumbledore DumbleWand) (wand HarryWand)
         (wand DumbleWand) (alive Dumbledore) (alive Harry)

Action: (send-for-help Dumbledore Harry Snape tower)
Location: Tower
Characters: Dumbledore, Harry
Dialogue: none
Effects: (person Snape) (conscious Snape)
         (has Snape SnapeWand) (wand SnapeWand)
         (alive Snape)

Action: (go Harry tower slytherin)
Location: Tower
```

```
Characters: Harry, Dumbledore
Dialogue: none
Effects: !(at Harry slytherin) (at Harry tower)

Action: (hide Harry)
Location: Tower
Characters: Harry, Dumbledore
Dialogue: none
Effects: (at Draco tower)

Action: (talk Dumbledore Draco tower)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: none

Action: (point-wand Draco Dumbledore DracoWand)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: (raised DracoWand)

Action: (watch Harry Draco)
Location: Tower
Characters: Harry, Dumbledore, Draco
Dialogue: none
Effects: none

Action: (point-wand Draco Dumbledore DracoWand)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: (raised DracoWand)

Action: (talk-down Dumbledore Draco)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: none

Action: (disarm Draco Dumbledore DracoWand DumbleWand)
Location: Tower
Characters: Draco, Dumbledore
Dialogue: none
Effects: none
```

```
Action: (hear Dumbledore Bellatrix)
Location: Tower
Characters: Draco, Dumbledore
Dialogue: none
Effects: (person Bellatrix) (has Bellatrix BellaWand)
         (wand BellaWand) (alive Bellatrix)

Action: (point-wand Draco Dumbledore DracoWand)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: (raised DracoWand)

Action: (talk Dumbledore Draco tower)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: none

Action: (plead Dumbledore Draco)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: none

Action: (hide Harry)
Location: Tower
Characters: Harry, Dumbledore
Dialogue: none
Effects: none

Action: (point-wand Draco Dumbledore DracoWand)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: (raised DracoWand)

Action: (go Bellatrix roomreqt tower)
Location: Tower
Characters: Dumbledore, Draco, Bellatrix
Dialogue: none
Effects: none

Action: (talk Dumbledore Bellatrix tower)
```

```
Location: Tower
Characters: Dumbledore, Draco, Bellatrix
Dialogue: none
Effects: none

Action: (raise-wand Harry HarryWand)
Location: Tower
Characters: Harry, Draco
Dialogue: none
Effects: none

Action: (point-wand Snape Harry SnapeWand)
Location: Tower
Characters: Snape, Harry, Draco
Dialogue: none
Effects: (raised SnapeWand) (hidden Snape)

Action: (lower-wand-from Harry HarryWand Draco)
Location: Tower
Characters: Snape, Harry, Draco
Dialogue: none
Effects: none

Action: (lower-wand-from Snape SnapeWand Harry)
Location: Tower
Characters: Snape, Harry, Draco
Dialogue: none
Effects: none

Action: (go Snape slytherin tower)
Location: Tower
Characters: Dumbledore, Draco, Snape, Bellatrix
Dialogue: none
Effects: !(hidden Snape)

Action: (lower-wand-from Draco DracoWand Dumbledore)
Location: Tower
Characters: Dumbledore, Draco, Snape, Bellatrix
Dialogue: none
Effects: none

Action: (watch Dumbledore Harry)
Location: Tower
Characters: Dumbledore, Draco, Snape, Bellatrix, Harry
Dialogue: none
```

```
Effects: none

Action: (plead Dumbledore Snape)
Location: Tower
Characters: Dumbledore, Draco, Snape, Bellatrix
Dialogue: none
Effects: none

Action: (point-wand Snape Dumbledore SnapeWand)
Location: Tower
Characters: Snape, Dumbledore, Draco, Bellatrix
Dialogue: none
Effects: (raised SnapeWand)

Action: (kill Snape Dumbledore SnapeWand)
Location: Tower
Characters: Snape, Dumbledore, Draco, Bellatrix
Dialogue: none
Effects: none
```

B.4.6 Harry Potter Alternate Version

```
Action: (go Draco slytherin tower)
Location: Slytherin
Characters: Draco
Dialogue: none
Plan: Draco:dracoplan
Effects: (person Draco) (place slytherin) (place tower)
         (hogwarts-place slytherin) (hogwarts-place tower)
         (conscious Draco) (has Draco DracoWand) !(at Draco tower)
         (wand DracoWand) (alive Draco) (at Bellatrix borgin)
         (passage-place borgin) (passage-place roomreqt)
         (place borgin) (place roomreqt) (hogwarts-place roomreqt)

Action: (apparate-with Dumbledore Harry hogsmeade tower)
Location: Tower
Characters: Harry, Dumbledore
Dialogue: none
Effects: (person Harry) (person Dumbledore) (place hogsmeade)
         (conscious Harry) (conscious Dumbledore)
         (headmaster Dumbledore) (has Harry HarryWand)
         (has Dumbledore DumbleWand) (wand HarryWand)
         (wand DumbleWand) (alive Dumbledore) (alive Harry)
```

```
Action: (send-for-help Dumbledore Harry Snape tower)
Location: Tower
Characters: Dumbledore, Harry
Dialogue: none
Effects: (person Snape) (conscious Snape)
         (has Snape SnapeWand) (wand SnapeWand)
         (alive Snape)

Action: (go Harry tower slytherin)
Location: Tower
Characters: Harry, Dumbledore
Dialogue: none
Effects: !(at Harry slytherin) (at Harry tower)

Action: (hide Harry)
Location: Tower
Characters: Harry, Dumbledore
Dialogue: none
Effects: (at Draco tower)

Action: (talk Dumbledore Draco tower)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: none

Action: (point-wand Draco Dumbledore DracoWand)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: (raised DracoWand)

Action: (watch Harry Draco)
Location: Tower
Characters: Harry, Dumbledore, Draco
Dialogue: none
Effects: none

Action: (point-wand Draco Dumbledore DracoWand)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: (raised DracoWand)
```

```
Action: (talk-down Dumbledore Draco)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: none

Action: (point-wand Dumbledore Draco DumbleWand)
Location: Tower
Characters: Draco, Dumbledore
Dialogue: none
Effects: (raised DumbleWand)

Action: (hear Dumbledore Bellatrix)
Location: Tower
Characters: Draco, Dumbledore
Dialogue: none
Effects: (person Bellatrix) (has Bellatrix BellaWand)
         (wand BellaWand) (alive Bellatrix)

Action: (point-wand Draco Dumbledore DracoWand)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: (raised DracoWand)

Action: (talk Dumbledore Draco tower)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: none

Action: (plead Dumbledore Draco)
Location: Tower
Characters: Dumbledore, Draco
Dialogue: none
Effects: none

Action: (hide Harry)
Location: Tower
Characters: Harry, Dumbledore
Dialogue: none
Effects: none

Action: (point-wand Draco Dumbledore DracoWand)
Location: Tower
```

```
Characters: Dumbledore, Draco
Dialogue: none
Effects: (raised DracoWand)

Action: (go Bellatrix roomreqt tower)
Location: Tower
Characters: Dumbledore, Draco, Bellatrix
Dialogue: none
Effects: none

Action: (talk Dumbledore Bellatrix tower)
Location: Tower
Characters: Dumbledore, Draco, Bellatrix
Dialogue: none
Effects: none

Action: (raise-wand Harry HarryWand)
Location: Tower
Characters: Harry, Draco
Dialogue: none
Effects: none

Action: (point-wand Snape Harry SnapeWand)
Location: Tower
Characters: Snape, Harry, Draco
Dialogue: none
Effects: (raised SnapeWand) (hidden Snape)

Action: (lower-wand-from Harry HarryWand Draco)
Location: Tower
Characters: Snape, Harry, Draco
Dialogue: none
Effects: none

Action: (lower-wand-from Snape SnapeWand Harry)
Location: Tower
Characters: Snape, Harry, Draco
Dialogue: none
Effects: none

Action: (go Snape slytherin tower)
Location: Tower
Characters: Dumbledore, Draco, Snape, Bellatrix
Dialogue: none
Effects: !(hidden Snape)
```

```
Action: (lower-wand-from Draco DracoWand Dumbledore)
Location: Tower
Characters: Dumbledore, Draco, Snape, Bellatrix
Dialogue: none
Effects: none

Action: (watch Dumbledore Harry)
Location: Tower
Characters: Dumbledore, Draco, Snape, Bellatrix, Harry
Dialogue: none
Effects: none

Action: (point-wand Dumbledore Snape DumbleWand)
Location: Tower
Characters: Dumbledore, Draco, Snape, Bellatrix
Dialogue: none
Effects: none

Action: (point-wand Snape Dumbledore SnapeWand)
Location: Tower
Characters: Snape, Dumbledore, Draco, Bellatrix
Dialogue: none
Effects: (raised SnapeWand)

Action: (kill Snape Dumbledore SnapeWand)
Location: Tower
Characters: Snape, Dumbledore, Draco, Bellatrix
Dialogue: none
Effects: none
```

REFERENCES

[1] Abbott, H. P., *The Cambridge Introduction to Narrative*. Cambridge University Press, 2008.

[2] Anderson, J. R., "A spreading activation theory of memory," *Journal of Verbal Learning and Verbal Behavior*, vol. 22, pp. 261–295, June 1983.

[3] Anderson, J. R., *Cognitive Psychology and its Implications*. New York: Worth Publishers, 6th ed., 2005.

[4] Aristotle, *The Poetics*. Buffalo, NY: Prometheus Books, 1992. T. Buckley translation. Original work published in 350 B.C.E.

[5] Bae, B. and Young, R. M., "A use of flashback and foreshadowing for surprise arousal in narrative using a Plan-Based approach," in *Proceedings of the International Conference on Digital Storytelling*, vol. 5334/2008, pp. 156–167, Springer, 2008.

[6] Bailey, P., "Searching for storiness: Story-generation from a reader's perspective," in *Narrative Intelligence: Papers from the AAAI Fall Symposium*, pp. 157–163, AAAI Press, 1999.

[7] Barber, H. and Kudenko, D., "Generation of dilemma-based interactive narratives with a changeable story goal," in *Proceedings of the 2nd International Conference on Intelligent Technologies for Interactive Entertainment*, (Cancun, Mexico), pp. 1–10, ICST, 2008.

[8] Bickmore, T., Schulman, D., and Yin, L., "Engagement vs. deceit: Virtual humans with human autobiographies," in *Proceedings of the 9th International Conference on Intelligent Virtual Agents*, vol. 5773 of *Lecture Notes in Computer Science*, (Amsterdam, The Netherlands), pp. 6–19, Springer Berlin Heidelberg, 2009.

[9] Bonet, B. and Geffner, H., "Planning as heuristic search: New results," in *Recent Advances in AI Planning: Proceedings of the 5th European Conference on Planning*, vol. 1809 of *Lecture Notes in Artificial Intelligence*, (Berlin), pp. 359–371, Springer, 1999.

[10] Bonet, B. and Geffner, H., "Planning as heuristic search," *Artificial Intelligence*, vol. 129, pp. 5–33, June 2001.

[11] Boujarwah, F. A., Abowd, G. D., and Arriaga, R. I., "Socially computed scripts to support social problem solving skills," in *Proceedings of the 2012 ACM*

Annual Conference on Human Factors in Computing Systems, (Austin, Texas, USA), pp. 1987–1996, ACM Press, 2012.

[12] Branigan, E., *Narrative Comprehension and Film.* New York: Routledge, 1992.

[13] Brewer, W. F. and Lichtenstein, E. H., "Stories are to entertain: A Structural-Affect theory of stories," *Journal of Pragmatics*, vol. 6, pp. 473–483, 1982.

[14] Bruner, J., "The narrative construction of reality," *Critical Inquiry*, vol. 18, no. 1, pp. 1–21, 1991.

[15] Campbell, M., "Casino royale," 2006.

[16] Cardona-Rivera, R. E., Cassell, B. A., Ware, S. G., and Young, R. M., "Indexter: A computational model of the event-indexing situation model for characterizing narratives," in *Proceedings of the 2012 Computational Models of Narrative Workshop*, 2012.

[17] Chambers, N. and Jurafksy, D., "Unsupervised learning of narrative event chains," in *Proceedings of the Forty-Sixth Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 789–797, Association for Computational Linguistics, 2008.

[18] Charmaz, K., *Constructing grounded theory: A practical guide through qualitative analysis.* Thousand Oaks, California, USA: Sage, 2006.

[19] Cheong, Y., *A Computational Model of Narrative Generation for Suspense.* PhD thesis, North Carolina State University, 2007.

[20] Cicchetti, D. V., "Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology," *Psychological Assessment*, vol. 6, pp. 284–290, Dec. 1994.

[21] Colton, S., Charnley, J., and Pease, A., "Computational creativity theory: The FACE and IDEA descriptive models," in *Proceedings of the Second International Conference on Computational Creativity*, (Mexico City), pp. 90–95, 2011.

[22] Comisky, P. and Bryant, J., "Factors involved in generating suspense," *Human Communication Research*, vol. 9, pp. 49–58, Sept. 1982.

[23] Cullingford, R. E., "SAM and micro SAM," in *Inside Computer Understanding* (Schank, R. and Riesbeck, C., eds.), Hillsdale, NJ: Erlbaum, 1981.

[24] Datta, R., Joshi, D., Li, J., and Wang, J. Z., "Studying aesthetics in photographic images using a computational approach," in *Computer Vision - ECCV 2006*, vol. 3953 of *Lecture Notes in Computer Science*, (Graz, Austria), Springer, May 2006.

[25] Fikes, R. E. and Nilsson, N. J., "STRIPS: a new approach to the application of theorem proving to problem solving," *Artificial Intelligence*, vol. 2, pp. 189–208, 1971.

[26] Fleiss, J. L., "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, vol. 76, pp. 378–382, Nov. 1971.

[27] Freytag, G., *The Technique of the Drama: An Exposition of Dramatic Composition and Art*. Johnston Reprint Corporation, 1968.

[28] Fujiki, T., Nanba, H., and Okumura, M., "Automatic acquisition of script knowledge from a text collection," in *Proceedings of the Tenth Conference on European chapter of the Association for Computational Linguistics*, (Budapest, Hungary), pp. 91–94, Association for Computational Linguistics, 2003.

[29] Gerrig, R. J., *Experiencing Narrative Worlds: On the Psychological Activities of Reading*. New Haven: Yale University Press, 1993.

[30] Gerrig, R. J. and Bernardo, A. B., "Readers as problem-solvers in the experience of suspense," *Poetics*, vol. 22, pp. 459–472, 1994.

[31] Gervás, P., "Computational approaches to storytelling and creativity," *AI Magazine*, vol. 30, no. 3, pp. 49–62, 2009.

[32] Gervás, P., Díaz-Agudo, B., Peinado, F., and Hervás, R., "Story plot generation based on CBR," *Knowledge-Based Systems*, vol. 18, pp. 235–242, Aug. 2005.

[33] Ghallab, M., Nau, D., and Traverso, P., *Automated planning: theory and practice*. San Francisco, CA, USA: Morgan Kaufmann, 2004.

[34] Graesser, A. C., Singer, M., and Trabasso, T., "Constructing inferences during narrative text comprehension.," *Psychological Review*, vol. 101, no. 3, pp. 371–395, 1994.

[35] Gratch, J., "Émile: Marshalling passions in training and education," in *Proceedings of the Fourth International Conference on Autonomous Agents*, (New York, NY), pp. 325–332, ACM, 2000.

[36] Hanks, S. and Weld, D. S., "A domain-independent algorithm for plan adaptation," *Journal of Artificial Intelligence Research*, vol. 2, pp. 319–360, 1995.

[37] Hitchcock, A., "Rear window," 1954.

[38] Hogg, C. M., *Learning Hierarchical Task Networks from Traces and Semantically Annotated Tasks*. PhD dissertation, Lehigh University, Bethlehem, PA, USA, Sept. 2011.

[39] Kasch, N. and Oates, T., "Mining script-like structures from the web," in *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, (Los Angeles, California, USA), pp. 34–42, Association for Computational Linguisitics, 2010.

[40] Kenny, P., Parsons, T., Gratch, J., and Rizzo, A., "Virtual humans for assisted health care," in *Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments*, (Athens, Greece), ACM, 2008.

[41] Keyder, E. and Geffner, H., "Heuristics for planning with action costs," in *Current Topics in Artificial Intelligence* (Borrajo, D., Castillo, L., and Corchado, J., eds.), vol. 4788 of *Lecture Notes in Computer Science*, pp. 140–149, Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-75271-4_15.

[42] Krippendorff, K., "Testing the reliability of content analysis data: What is involved and why," in *The Content Analysis Reader* (Krippendorff, K. and Bock, M. A., eds.), pp. 350–357, Thousand Oaks, CA, USA: Sage Publications, 2009.

[43] Landis, J. R. and Koch, G. G., "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, pp. 159–174, Mar. 1977.

[44] Law, E. and von Ahn, L., *Human Computation*. Morgan & Claypool, 2011.

[45] Lebowitz, M., "Story-telling as planning and learning," *Poetics*, vol. 14, pp. 483–502, 1985.

[46] Li, B., Lee-Urban, S., Johnston, G., and Riedl, M. O., "Story generation with crowdsourced plot graphs," in *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, (Bellevue, Washington, USA), AAAI, 2013.

[47] Li, B. and Riedl, M. O., "An offline planning approach to game plotline adaptation," in *Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp. 45–50, AAAI Press, 2010.

[48] MacLeod, C. and Campbell, L., "Memory accessibility and probability judgments: An experimental evaluation of the availability heuristic," *Journal of Personality and Social Psychology*, vol. 63, pp. 890–902, Dec. 1992.

[49] Marsella, S. C. and Gratch, J., "EMA: a process model of appraisal dynamics," *Cognitive Systems Research*, vol. 10, pp. 70–90, Mar. 2009.

[50] Mateas, M., *Interactive Drama, Art and Artificial Intelligence*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, Dec. 2002.

[51] Mawhorter, P. and Mateas, M., "Reader-model-based story generation," in *Proceedings of the 9th AAAI Conference on Artificial Intelligence for Interactive Digital Entertainment*, (Boston, Massachusetts), AAAI, 2013.

[52] Meehan, J., "TALE-SPIN," in *Inside Computer Understanding* (Schank, R. C. and Riesbeck, C. K., eds.), pp. 197–226, Hillsdale, NJ: Lawrence Erlbaum Associates, 1981.

[53] Miles, M. B. and Huberman, A. M., *Qualitative data analysis*. Thousand Oaks, California, USA: Sage, 2 ed., 1994.

[54] Neal Reilly, W. S., *Believable Social and Emotional Agents*. PhD dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 1996.

[55] Nelson, M. J. and Mateas, M., "Search-based drama management in the interactive fiction anchorhead," in *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference*, (Marina del Rey, California), 2005.

[56] Nelson, M. J., Mateas, M., Roberts, D. L., and Isbell, C. L., "Declarative optimization-based drama management in interactive fiction," *IEEE Computer Graphics and Applications*, vol. 26, pp. 32–41, June 2006.

[57] Niehaus, J., *Cognitive Models of Discourse Comprehension for Narrative Generation*. PhD dissertation, North Carolina State University, 2009.

[58] Orkin, J. D., *Learning Plan Networks in Conversational Video Games*. Masters thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, Aug. 2007.

[59] Ortony, A., Clore, G. L., and Collins, A., *The Cognitive Structure of Emotions*. Cambridge, UK: Cambridge University Press, 1988.

[60] Pérez y Pérez, R. and Sharples, M., "MEXICA: a computational model ofa cognitive account of creative writing," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 13, no. 2, pp. 119–139, 2001.

[61] Porteous, J. and Cavazza, M., "Controlling narrative generation with planning trajectories," in *Proceedings of the 2nd International Conference on Interactive Digital Storytelling*, pp. 234–245, 2009.

[62] Porteous, J., Teutenberg, J., Pizzi, D., and Cavazza, M., "Visual programming of plan dynamics using constraints and landmarks," in *Proceedings of the 21st International Conference on Automated Planning and Scheduling*, (Freiburg, Germany), pp. 186–193, AAAI Press, June 2011.

[63] Prince, G., *Dictionary of Narratology*. Lincoln, Nebraska, USA: University of Nebraska Press, 2003.

[64] Reich, Y., "A model of aesthetic judgment in design," *Artificial Intelligence in Engineering*, vol. 8, no. 2, pp. 141–153, 1993.

[65] Riedl, M. O. and Bulitko, V., "Interactive narrative: An intelligent systems approach," *AI Magazine*, vol. 34, no. 1, 2013.

[66] Riedl, M. O. and Young, R. M., "Narrative planning: Balancing plot and character," *Journal of Artificial Intelligence Research*, vol. 39, pp. 217–268, Sept. 2010.

[67] Saldaña, J., *The Coding Manual for Qualitative Researchers*. Los Angeles: Sage Publications, 2009.

[68] Schank, R. C. and Abelson, R., *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*, vol. 2. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1977.

[69] Strauss, A. L., *Qualitative analysis for social scientists*. Cambridge, UK: Cambridge University Press, 1987.

[70] Szilas, N., "A computational model of an intelligent narrator for interactive narratives," *Applied Artificial Intelligence*, vol. 21, pp. 753–801, 2007.

[71] Szilas, N., "Requirements for computational models of interactive narrative," in *Proceedings of the AAAI Fall Symposium on Computational Models of Narrative*, AAAI, 2010.

[72] Tan, E. S., *Emotion and the Structure of Narrative Film: Film as an Emotion Machine*. New York: Routledge, 1996.

[73] Turner, S. R., *Minstrel: A Computer Model of Creativity and Storytelling*. PhD dissertation, University of California, Los Angeles, 1993.

[74] Weyhrauch, P., *Guiding Interactive Drama*. Ph.D., Carnegie Mellon University, Pittsburgh, PA, USA, 1997.

[75] Wilensky, R., "PAM and micro PAM," in *Inside Computer Understanding* (Schank, R. and Riesbeck, C., eds.), Hillsdale, NJ: Erlbaum, 1981.

[76] Yates, D., "Harry potter and the half-blood prince," 2009.

[77] Yu, H. and Riedl, M. O., "Personalized interactive narratives via sequential recommendation of plot points," *Transactions on Computational Intelligence and Artificial Intelligence in Games*, 2013.

[78] Zagalo, N., Barker, A., and Branco, V., "Story reaction structures to emotion detection," in *Proceedings of the 1st ACM Workshop on Story Representation, Mechanism and Context*, (New York), pp. 33–38, ACM, 2004.

[79] Zillmann, D., "The logic of suspense and mystery," in *Responding to the Screen: Reception and Reaction Processes* (Bryant, J. and Zillmann, D., eds.), pp. 281–303, Hillsdale, NJ: Lawrence Erlbaum Associates, 1991.

[80] Zillmann, D., "The psychology of suspense in dramatic exposition," in *Suspense: Conceptualizations, Theoretical Analyses, and Empirical Explorations* (Vorderer, P., Wulff, H. J., and Friedrichsen, M., eds.), pp. 199–231, Mahwah, NJ: Lawrence Erlbaum Associates, 1996.

[81] Zwaan, R. A., Langston, M. C., and Graesser, A. C., "The construction of situation models in narrative comprehension: An event-indexing model," *Psychological Science*, vol. 6, pp. 292–297, Sept. 1995.

VITA

Brian C. O'Neill was born in New York, NY, USA on August 27, 1985, to Patrick and Jennifer O'Neill. He earned his Bachelor of Science in Computer Science from Saint Joseph's University in May 2007, where he graduated magna cum laude. While at the Georgia Institute of Technology, he was a Research Assistant in the Contextualized Support for Learning Lab from 2007-2008, working on an integrated development environment for non-major computer science students; a Research Assistant in the Entertainment Intelligence Lab (also known as the Intelligent Narrative Computing Group) from 2008-2013, studying computational creativity, story and discourse generation, and computational models of readers; a summer Instructor, teaching an upper-level elective in Game AI in 2012; and the president of the College of Computing Graduate Student Council in 2012-2013. He earned his Masters in Computer Science from Georgia Tech in December 2010. He will start as an Assistant Professor in the Department of Computer Science and Information Technology at Western New England University in January 2014.