ABSTRACT

RIEDL, MARK OWEN. Narrative Planning: Balancing Plot and Character. (Under the direction of R. Michael Young.)

The ability to generate narrative is of importance to computer systems that wish to use story effectively for a wide range of contexts ranging from entertainment to training and education. The typical approach for incorporating narrative into a computer system is for system builders to script the narrative features at *design time*. A central limitation of this pre-scripting approach is its lack of flexibility -- such systems cannot adapt the story to the user's interests, preferences, or abilities. The alternative approach is for the computer systems themselves to generate narrative that is fully adapted to the user at *run time*.

A central challenge for systems that generate their own narrative elements is to create narratives that are readily understood as such by their users. I define two properties of narrative – *plot coherence* and *character believability* – which play a role in the success of a narrative in terms of the ability of the narrative's audience to comprehend its structure. Plot coherence is the perception by the audience that the main events of a story have meaning and relevance to the outcome of the story. Character believability is the perception by the audience that the actions performed by characters are motivated by their beliefs, desires, and traits.

In this dissertation, I explore the use of search-based planning as a technique for generating stories that demonstrate both strong plot coherence and strong character believability. To that end, the dissertation makes three central contributions. First, I describe an extension to search-based planning that reasons about character intentions by identifying

possible character goals that explain their actions in a plan and creates plan structure that explains why those characters commit to their goals. Second, I describe how a character personality model can be incorporated into planning in a way that guides the planner to choose consistent character behavior without strictly preventing characters from acting "out of character" when necessary. Finally, I present an open-world planning algorithm that extends the capabilities of conventional planning algorithms in order to support a process of story creation modeled after the process of dramatic authoring used by human authors. This open-world planning approach enables a story planner not only to search for a sequence of character actions to achieve a set of goals, but also to search for a possible world in which the story can effectively be set.

The planning algorithms presented in this dissertation are used within a narrative generation system called *Fabulist*. Fabulist generates a story as a sequence of character actions and then recounts the story by first generating a discourse plan that specifies how the story content should be told and then realizing the discourse plan in a storytelling medium. I present the results of an empirical evaluation that demonstrates that narratives generated by Fabulist have strong plot coherence and strong character believability. The results clearly indicate how a planning approach to narrative generation that reasons about plot coherence and character believability can improve the audience's comprehension of plot and character.

NARRATIVE GENERATION: BALANCING PLOT AND CHARACTER

by

MARK OWEN RIEDL

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

COMPUTER SCIENCE

Raleigh

2004

APPROVED BY:

Chair of advisory committee

To my parents.

Biography

The story of Mark Riedl's academic career has always been about making computers interact with human users more intelligently. As an undergraduate, Mark supplemented the Computer Science curriculum at NC State University with courses in cognitive psychology and earned his Bachelor's Degree in Computer Science with a minor is Psychology in 1999. Mark began his graduate study in 1999 under the guidance of Dr. Robert St. Amant doing research that applied artificial intelligence to human-computer interaction. Mark earned his Master's Degree in Computer Science in 2001 with his thesis, *A Computational Model of Navigation in Social Environments*, which presented a computational account for human social navigation behavior. In order to push the limits on intelligent human-computer interaction, Mark joined the Liquid Narrative group run by Dr. R. Michael Young. As a member of the Liquid Narrative group, Mark researched ways in which computer systems can use storytelling as a way of guiding their interaction with interactive, human users.

Acknowledgements

Foremost, I would like to thank my committee members, Michael Capps, Jon Doyle, James Lester, and Brad Mehlenbacher for their guidance, concern, and enthusiasm.

Throughout my time as a graduate student, I have had several advisors who I am indebted to. Robert St. Amant taught me to be a scientist. Although our interests diverged, I don't think I would be where I am today without him. Michael Young for his willingness to continuously bend over backwards to ensure that I had every opportunity to be successful.

Many people have been generous and helpful over the years. In the larger world, I wish to thank those professional colleagues who have invited me to be their peers: William Bares, Marc Cavazza, Fred Charles, and Lewis Johnson. At NC State University, I have had the support and confidence of a number of faculty: James Lester, Carol Miller, Matthias Stallmann, Robert St. Amant, and Michael Young. A large number of fellow graduate students have been there for me: Charles Callaway, Yuna Cheong, Pankaj Chopra, David Christian, Brent Dennis, Justin Harris, Arnav Jhala, Sarat Kocherlakota, Seung Lee, Brad Mott, Brian Shiver, Laura Tateotian, Tommy Vernieri, Joe Winegarden, Alex Woods, and the rest of the Mimite hoard (too numerous to name, but you know who you are).

I would also like to thank my parents, Dr. Richard Riedl and Theresa Riedl, for encouraging me to pursue my interests and overcome my weaknesses and for all their support, advice, and unconditional love. I would like to express my gratitude to my closest friends for their understanding and patience. Finally, I would like to thank my best friend of all, Jennifer King, for her patience, encouragement, emotional support, and willingness to not only stand by me, but to marry me.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Narrative as entertainment, in the form of oral, written, or visual stories, plays a central role in our social and leisure lives. Narrative is also used in education and training contexts to motivate and to illustrate. One reason for this is that cognitive structures we use to understand the world around us are similar to the cognitive structures we use to understand narratives (Bruner, 1990). In order to act effectively in the world, one must understand the world one is situated in. Our understanding of the world is achieved by "constructing reality" as a sequence of related events from our senses (Bruner, 1991). The cognitive process of structuring related events enables one to extract meaning from changes in the world and to make inferences about the future. Essentially, we understand the world by telling ourselves stories about how we have changed the world and witnessed the world change. This autobiographical representation of memory and thought has evolved from the social nature of our species because stories are a highly efficient and natural way to communicate (Dautenhahn, 2003). Whereas we tend to understand inanimate objects through cause and effect, we attempt to understand the intentional behavior of others through a sophisticated process of interpretation with narrative at its core (Bruner, 1990; Sengers, 2000a).

The goal of cognitive scientists is to understand and predict human behavior as a process of conscious decision-making. One goal of artificial intelligence research is to computationally replicate human cognitive ability as a way of solving applied problems or testing theories of cognition. Given the importance of narrative in human experience, it was perhaps inevitable that cognitive science and artificial intelligence tackle the problem of narrative understanding and generation. The research agenda of Roger Schank and his students in the 1970's and early 1980's was to explore the knowledge structures and processes that humans used to understand the meaning of natural language. Since the meaning of sentences cannot always be determined without a larger context (Schank, 1977), Schank and his team turned to computational models of understanding – and then later of generating – stories. Later, artificial intelligence research took a more applied approach to narrative generation, focusing on the entertainment and educational goal of building *interactive narrative systems*. An interactive narrative system implements a virtual world in which a computer user can enter the world as a character in an unfolding narrative. Recently, Blair and Meyer (1997) coined the term "narrative intelligence" to refer to the ability – human or computer – to organize experience into narrative. A computer system that uses a narrative approach to entertainment or education will benefit from the ability to reason about narrative intelligence because the system can structure its narrative in ways that afford understanding by the user.

Computational systems that reason about narrative intelligence are able to interact with human users in a natural way because they understand collaborative contexts as emerging narrative and are able to express themselves through storytelling. The standard approach to incorporating storytelling into a computer system, however, is to script a story at design time. That is, the system designers determine ahead of time what the story should be and hard-code the story into the system. For a computer system to use a scripted story means that the ability of the system to adapt to the user's preferences and abilities is limited. The story scripted into a system may not completely engage the user's interests or may be too challenging for the user to follow. Furthermore, if stories are scripted at design time, a system can only have a limited number of stories it can present to the user. In entertainment applications, a limited number of stories or a limited number of permutations of a single story in a computer game results limited replay value of that game. In educational and training applications, a limited number of stories or a limited number of permutations of a single story results in a system that cannot fully cater to the student's needs and abilities. The alternative approach is to generate stories either dynamically or on a per-session basis (one story per time the system is engaged). Narrative generation is a process that involves the selection of narrative content (the events that will be presented to an audience), ordering of narrative content and presentation through discourse of narrative content. A system that can generate stories is capable of adapting narrative to the user's preferences and abilities, has expanded "replay value" and is capable of interacting with the user in ways that were not initially envisioned by system designers.

1.1. Motivation

The applications that can benefit from storytelling are many and varied. Storytelling has long been associated with computer games; story provides scaffolding as motivation for why the user should suspend her disbelief and play the game (Juul, 2001). Many computer games tell a story to the user as the user manipulates a character through levels but the plot of the story is typically scripted. The only variation in story is in the moment-to-moment sequencing of actions that the user performs. Major developments in the story that do not involve the life or death of the protagonist are shown through pre-scripted cinematic sequences – cut-scenes – in which the user is temporarily relieved of control while the storyline unfolds.

Education and training systems have also successfully applied storytelling. The Victec project (Sobral, Machado, & Paiva, 2003a; Sobral, Machado, & Paiva, 2003b) uses stories to illustrate effective strategies for children to handle bullying situations. Carmen's Bright Ideas (Marsella, Johnson, & LaBore, 2000) uses a story to teach parents of pediatric cancer patients coping strategies. The ICT Leaders project (Gordon et al., 2004) uses story to involve a student in situations where leadership decisions must be made to complete a task.

There are two fundamental types of narratives used in computer games and education and training applications: linear narrative and branching narrative. Linear narrative is the traditional form of narrative in which a sequence of events is narrated from beginning to ending without variation or possibility of a user altering the way in which the story unfolds or ends. Computer games typically employ linear narratives although the story structure is partitioned into interactive portions – levels – and cut-scenes. Even though the user has a certain degree of control during a level, the only outcome is successful completion of some objective (usually killing all the enemies in an area) or failure, in which case the user must start the level over. All users experience the same story and each user will experience the same story during successive sessions.

Some computer games use branching narrative in which there are many points in the story at where some action or decision made by the user alters the way in which a narrative unfolds or ends. Educational and training systems typically use branching narrative so that students can apply their skills or test their understanding. Branching narratives (e.g. Kelso, Weyhrauch, & Bates, 1993; Galyean, 1995; Silva, Raimundo, & Paiva, 2003; Gordon et al., 2004) are typically represented as directed graphs in which each node represents a linear, scripted scene followed by a decision point. Arcs between nodes represent decisions that can be made by the user. Even though a branching narrative may introduce variability into the experience a user has with a storytelling system, the variability is scripted into the system at design time and is thus limited by the system designer's anticipation of the user's needs or preferences. All users will be presented with the same set of decision points and each user will experience the same story if she makes the same set of choices.

1.1.1. Linear Narrative

Any system that requires a script of action for one or more agents to follow can benefit from the ability to generate linear narrative. The range of applications is broad. The applications that most obviously can benefit from the ability to generate linear narrative are those that produce scripts, stories, or movies for the purpose of entertaining an audience. While it is not likely that computer applications will replace human authors of stories and movie and TV scripts any time soon, story generation systems can be tools for authors. For example, a defense-related application can be asked to rapidly generate a number of plausible narratives describing how a particular world state might be achieved (e.g. the outcome of an emergency rescue operation). For entertainment, a system that can produce stories on request has similar potential.

Computer games can benefit from being able to dynamically generate cut-scenes. Computer games typically involve episodes of interactivity separated by pre-scripted cut scenes. The interleaving of levels and cut-scenes requires that a very specific condition of the world must be met at the end of each interactive episode so that there is a coherent transition from interactivity to cut-scene animation. Typically, an interactive episode ends with the user moving her embodied character through a doorway so that the cut scene can begin with an animation of that character, no longer under the control of the user, walking through the doorway. A computer game that dynamically generates cut-scenes could generate coherent cut-scenes in levels with multiple ending conditions, including end conditions that the system designers did not initially foresee.

In the contexts of education and training, systems that can dynamically generate narratives can illustrate learning principles through stories that are tailored to individual students' interests and abilities. Students respond better to systems that account for individual difference between users such as age, sex, and interest in particular themes. For example, in a system that teaches students about history, the context surrounding an historical event can be taught through fictional accounts of the people that lived in a particular time (Riedl & Young, 2004). Educational systems can also teach by generating Aesop-like fables to illustrate principles and morals that are directly relevant to a particular context. Training systems can generate stories to demonstrate how to achieve some particular goal or even to project possible outcomes of student decisions.

1.1.2. Branching Narrative

Branching narratives enable system users to influence the way in which a narrative unfolds or ends. At set points in branching narratives, the user is able to make a choice or perform an action that selects one of a set of alternative continuations of the storyline. Branching narratives are typically represented as a directed graph in which a node in the graph represents a linear story sequence followed by a decision point. The computational complexity of scripting story graphs at design time means that story graphs have either a low branching factor or a limited number of decision points (Bruckman, 1990). An interactive narrative system that represents story as a graph can be considered a story generation system because the story – the end result of a user's session – is not fixed at design time. The system, however, is merely piecing together pre-scripted sequences based on feedback from the user. The user is constrained to the structure of the branching story graph constructed by the system designer. Users that make the same choices at each decision point will have identical experiences with the system. If a user were to make the same decisions during two consecutive sessions with the system, her experience would be the same.

Interactive narrative systems not only have to consider the quality of the storytelling experience, but must balance the coherence of a story against the amount of control afforded the user (Riedl, Saretto, & Young, 2003). Providing more control to the user means increasing the possibility that the story will lose coherence and become less understandable. For that reason, branching narratives in interactive narrative systems are typically constructed at design time. One way to generate a branching narrative that balances coherence and control is to generate a linear narrative and then consider all the ways in which a user – playing a character situated in the story world – can interact with the objects and other characters in the world (Young & Riedl, 2003; Riedl, Saretto, & Young, 2003; Young et al, 2004). For every action that the user makes that threatens to deviate too severely from the linear story proposed by the system, the system dynamically generates an alternative storyline starting from the point of deviation. An alternative linear story structure exists for every potentially threatening user action. The combination of alternative linear stories is at least as expressive as an acyclic story graph. See (Riedl, 2004) for further discussion, including a proof of the expressivity claim.

An interactive narrative system that can generate a branching story graph dynamically can increase the degree of interactivity afforded to a user playing a character in the story world. Instead of being limited to periodic decision points, the user would be free in such a system to perform actions at any point throughout the session and the system would respond accordingly by adapting the story to the user's preferences and abilities. Computer games that are able to dynamically generate narrative structures can provide more opportunities for the user to influence the way in which a game's plot unfolds. Educational systems in which the user has the flexibility to experiment learns through interacting with a virtual world and the characters that inhabit it instead of being restricted to pre-specified points of interaction. Narrative plays such a central role in memory by providing an organizing structure for new experiences and knowledge (Mandler, 1984). A curriculum based around narrative techniques can increase learning effectiveness and motivation (Mott, 1999).

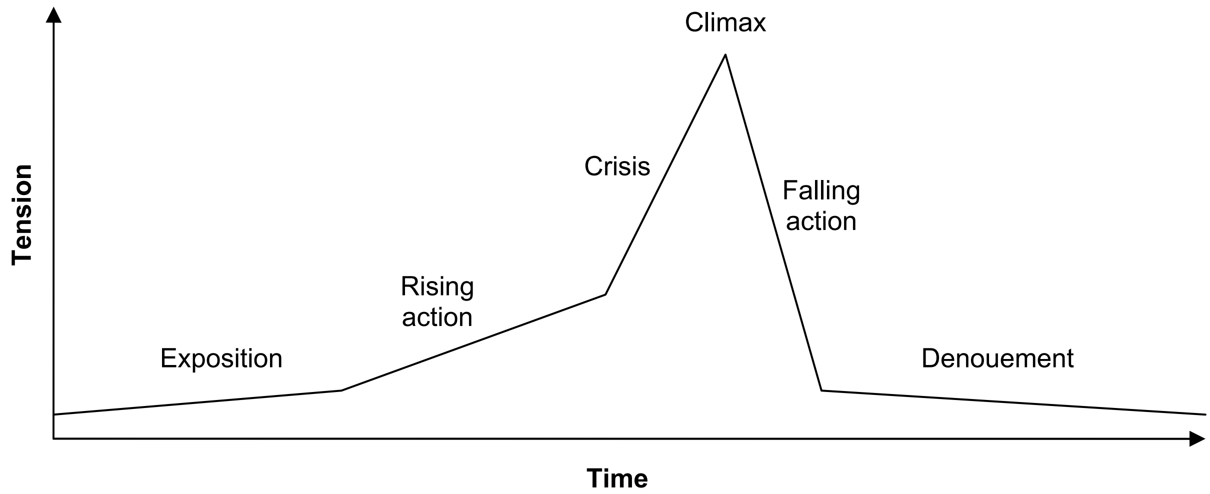

Figure 1.1. Aristotle's dramatic arc.

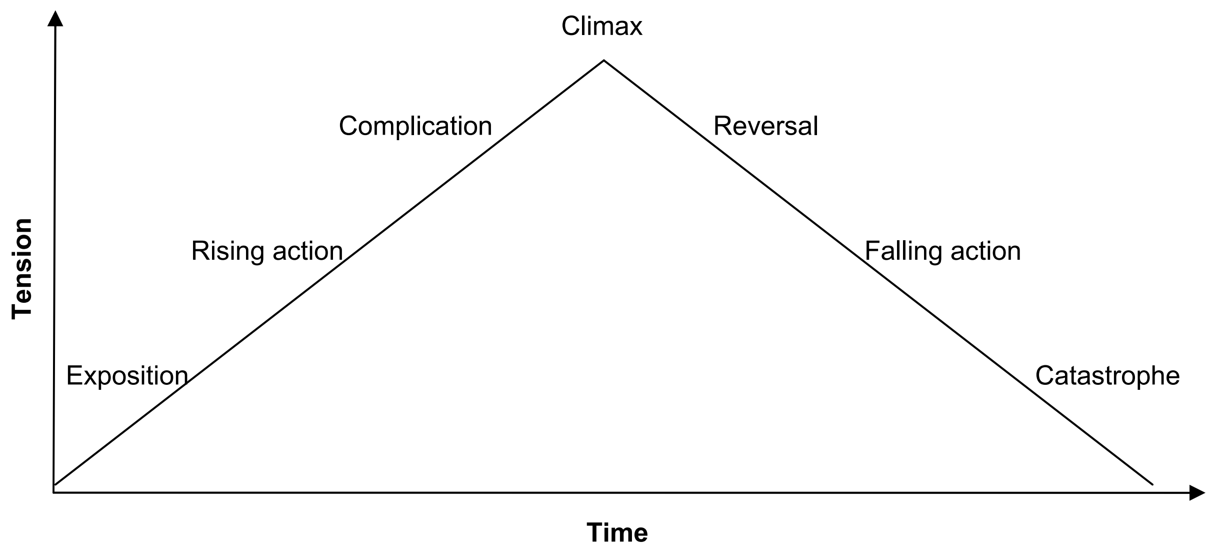

Figure 1.2. Freytag's pyramid.

1.2. Narrative

Narrative is associated with storytelling but the two terms are not equivalent. Narrative is the recounting of a sequence of events (Prince, 1987). To distinguish narrative from a recounting of a laundry list of random events, those who study narrative often consider narrative to have a "continuant subject and constitute a whole" (Prince, 1987, p. 58). For a narrative to have a continuant subject and constitute a whole, the events described in the narrative have a single point or relate to a single communicative goal (Chatman, 1993). One can, however distinguish between narratives that "tell a story" and narratives that do not (Herman, 2002). A narrative that tells a story has certain properties that one comes to expect. In particular, stories have plot. A plot is the list of main incidents in a narrative that make up an outline of the most important events in the story that is told. Plot adheres to a particular pattern that describes the order of events in time (Chatman, 1993). Aristotle suggests that narrative is patterned in such a way that the tension the audience feels in response to the story changes in a predictable way over time as the story is told. Aristotle's dramatic arc, as shown in Figure 1.1, is one common model of story structure. In a story with more than one act, the dramatic arc of each successive act starts at a higher level of tension and reaches a higher climax than the previous act. Another pattern of events in a narrative used explicitly for the genre of the tragedy is Freytag's Pyramid shown in Figure 1.2. The fact that narrative is a recounting of a sequence of events implies that all narratives have a narrator – an agent who tells a story to an audience (Bal, 1997). The narrator is not necessarily equivalent to the author of the narrative and can be an agent that is external to the story world or part of the story world as a character.

1.2.1. Narratology

Narratology is the theory of narratives that "tell a story" (Bal, 1997). Recently, narratology has been extended to apply to additional forms of audiovisual media such as movies (Brannigan, 1992). Narratology is the study of the structural elements of narrative media in order to devise a *narrative system* that generalizes the form of all narratives. Narratologists divide a narrative into three levels of interpretation: text or media, sjužet, and fabula. The levels of interpretation are shown in Table 1.1. At the lowest level of interpretation is the fabula. The fabula of a narrative consists of an enumeration of the events that occur in the

Table 1.1. The layers of interpretation of narrative.

Level	Meaning
Media/text	The surface-level communicative form that the narration takes.
Sjužet	The discourse that relates some of the events in the fabula from narrator to audience.
Fabula	The events in the story world that make up the content of the narrated material.

story world between the time that the story begins and the time that the story ends. The events in the fabula are ordered chronologically, in the order that they occur in the story world. The fabula is relatively complete; it contains events that are not necessarily part of the discourse of telling the story to an audience.

The sjužet is the discourse layer of the narrative. The sjužet consists of a subset of the events in the fabula that are actually told to the audience. Some events from the fabula are omitted from the discourse because they are irrelevant to the intentions of the narrator. Some events are omitted because they can be inferred. For example, a narrative about a businessman who travels from Los Angeles to Chicago by airplane might have a description of the businessman's arrival at the Los Angeles airport followed by a description of the businessman's departure from the Chicago airport. The description of the sequence of events involving the businessman's assumedly stereotypical experiences in both airports and on the airplane is unnecessary since the author assumes that his target audience has a schema which describes a typical air travel experience. Had any deviation from the schema been significant to the outcome of the story or the narrator's communicative intentions, those exceptional events might have been included in the discourse.

At the highest layer of interpretation in the narrative structure is the text or media. At this layer, the events of the sjužet are described to the audience in some communicative medium. If the narrative is in written or spoken format, this layer includes the natural language used to describe the events of the sjužet. If the narrative is cinematic, this layer includes the actions that actors perform and the camera operations that capture those actions.

The separation of a narrative into structural layers is an artificial distinction. The author of a narrative does not necessarily consider the fabula, sjužet, and media separately during the process of creating a narrative. However, the distinction between layers of interpretation is useful for analysis of narratives. Issues of *story* as represented by the fabula are clearly separated from issues of *storytelling* as represented by sjužet and media. From an engineering standpoint, this separation allows one to concentrate on story generation without becoming entangled in reasoning about discourse. Furthermore, once a fabula is in existence in whatever form, one can now reason about discourse and media for telling of the story independent from the generative issues relevant to the fabula.

1.2.2. Narrative Terminology

The following are narratological terms that are applicable to research in narrative generation. The definitions are taken from Prince (1987).

> **Definition 1.1 (Narrative).** *A narrative is a recounting of one or more real or fictitious events communicated by one or more narrators to one or more narratees.*

It is assumed that narrative, as a form of discourse, has a purpose. The purpose of a narrative is often referred to as the narrative's point. The plot of a narrative is the main incidents of a narrative that outline the way in which the point of a narrative. Plot is typically structured with regard to the way in which the events of a story impact the audience. It is possible for a narrative not to have a plot, although the narratives that are most often considered "stories" are those that have a plot.

> **Definition 1.2 (Plot).** *The plot of a narrative is an outline of the main incidents that constitute a structure characterizable in terms of Freytag's pyramid or Aristotle's dramatic arc.*

Narratologists are interested in the structure of narratives. The structure of a narrative can be analyzed on many different levels, as discussed in the previous section. The ordering of events is particularly important to an audience's understanding of a narrative. The temporal ordering of events, however, can differ depending on whether one is analyzing the narrative on the level of fabula or sjužet.

> **Definition 1.3 (Fabula).** *The fabula of a narrative is the set of narrated situations and events in their chronological sequence.*

> **Definition 1.4 (Sjužet).** *The sjužet of a narrative is the set of narrated events in the order of their presentation to the reader.*

Definition 1.5 (Story). *The story of a narrative is the content of a narrative as opposed to the expression of the narrative.*

For the purpose of the discussion in this dissertation, story is equivalent to fabula. The content (story) of a narrative is made up of events. These events can be further categorized depending on the agent that causes them to occur.

Definition 1.6 (Actor). *An actor is an entity consisting entirely of functions and attributes. A function is an operation that the actor can perform to change the state of the world. An attribute is a trait possessed by the actor.*

Definition 1.7 (Act). *An act is a type of event that can occur in a narrative that is characterized by a change in the state of the world that is brought about by some actor.*

Defintion 1.8 (Happening). *A happening is a type of event that can occur in a narrative that is characterized by a change in the state of the world that is* not *brought about by some actor.*

Acts and happenings in the story world together make up the events of a narrative. The artificial intelligence community uses the terms *action* and *operation* to mean intentional behavior that is performed in order to change the state of the world. These terms are interchangeable with the narratological term, *act*. Consequently, actions and operations are events. I use the terms *event*, *action*, and *operation* interchangeably throughout this dissertation. I use the terms *acts* and *happenings* when I need to make a clear distinction between intended and unintended events.

The term actor is a generic term that refers to anything that has the ability to change the state of the story world. However, it is common to refer to characters instead of actors.

Definition 1.9 (Character). *A character is an actor endowed with anthropomorphic traits and engaged in anthropomorphic actions.*

In my work, I do not draw a distinction between actors and characters. I use the two terms interchangeably. The artificial intelligence community uses the term *agent* to refer to an entity that is endowed with the ability to change the state of the world. An agent is therefore an actor.

1.3. Problem Statement

A narrative is the recounting of a sequence of events. As mentioned above, a narrative is about something and is detailed enough for one to understand the relevance of events towards that something[1]. All narratives describe how a certain state of affairs in the story world is transformed into some resulting state. One event leads to another event and so on until the inevitable conclusion is reached. The causality of events is an inherent property of narratives that ensures a whole and continuant subject (Chatman, 1993). However, even if a narrative is complete and continuous, some narratives are considered "stories" while others are not. Narratives that can be considered to "tell a story" have a quality of tellability – meaning that they have some quality that makes them something worth being told – while other narratives do not. Narratives that are considered tellable are those in which the events in the narrative are the intentional consequence of the beliefs, desires, plans, and goals of characters in the story world (Herman, 2002).

The causality of events and intentionality of characters that make certain stories more appealing than others is not arbitrary. Cognitive psychologists have determined that the ability of an audience to comprehend a narrative is strongly correlated with the causal structure of the story (Trabasso & Sperry, 1985; van den Broek, 1988; Graesser, Lang, & Roberts, 1991) and the attribution of intentions to the characters that are participants in the events (Graesser, Lang, & Roberts, 1991; Gerrig, 1993). Story comprehension requires the audience (e.g. reader, hearer, viewer) to perceive the causal connectedness of story events and to infer intentionality of characters. I define two properties of story that must be perceived by an audience for the story to be considered successful.

> **Definition 1.10 (Plot coherence).** *Plot coherence is the perception that the main events of a story are causally relevant to the outcome of the story.*

[1] This corresponds to Grice's maxims of *quantity* and *relation* (Grice, 1975). The maxim of quantity states that one should (a) make one's contribution as informative as necessary, and (b) not make one's contribution to the conversation more informative than necessary. The maxim of relation states that one should make one's contribution relevant.

Definition 1.11 (Character believability). *Character believability is the perception that the events of a story are reasonably motivated by the beliefs, desires, and goals of the characters that participate in the events.*

Plot coherence is the way in which the main events that make up the plot of a story appear to form a chain that necessarily terminates in the outcome of the story. Plot coherence only applies to the main events; the less significant events that lead to the main plot points do not need to be relevant to the outcome of the story, although in many cases they will, since they lead up to the main plot points. It is not enough that a narrative be merely coherent. A coherent narrative is one in which one event follows from another (Meehan, 1976; Lebowitz, 1985). Any narrative whose fabula is not a random list of unrelated events is coherent. Plot coherence refines the notion of coherence by suggesting not only that events are causally relevant to each other, but that the chain of causality eventually leads to the satisfaction of a point. Plot coherence is coherence that is goal-driven, reflecting the fact that an audience expects a story to have some point (Wilensky, 1982). When a story has significant events that do not appear to be relevant to the way the story ends, the direction of the story's plot is not completely focused on the outcome[2]. As a result, the audience may make incorrect inferences about the direction of the plot and in general have a lower level of story comprehension (Gerrig, 1993).

Character believability refers to the numerous elements that allow a character to achieve the "illusion of life" (Bates, 1994). It does not refer to a character that is honest or reliable, but to a character that convinces us to suspend our disbelief in the possibility that that character exists in some story world. Character believability is partially dependent on the idiosyncrasies of a character's appearance and physical movements. Physical appearance is very important in visual media such as animated film (Thomas & Johnson, 1981). The description of appearance is also found in written and spoken presentations. Equally important is the way in which the internal attributes of a character such as personality, emotion, desires, and intentions manifest themselves through the decisions the character makes and the behaviors the character performs (Thomas & Johnson, 1981; Bates, 1994;

[2] If a story is not plot coherent, then it has main events that are not causally relevant to the outcome of the story. Such a narrative violates Grice's maxim of quantity (the contribution is more informative than necessary).

Loyall, 1997). The definition of character believability places emphasis on the goal-oriented nature of characters. Goal-oriented behavior is a primary requirement for believability (Charles, 2003).

The purpose of the research presented in this dissertation is to investigate a technique for automatically generating narratives whose stories are plot coherent and character believable. My technique for generating narratives is implemented in the *Fabulist* system which treats the process of narrative generation as a process of generating a fabula and then transforming the fabula into sjužet and media. A fabula is constructed by a planner that emphasizes plot coherence and character believability. By reasoning about plot coherence and character believability, the fabula planner is able to construct stories whose structures facilitate audience understanding of causality and character intentionality. The core contribution of this research is a planning algorithm that is capable of generating stories that are structured so as to enhance the audience's perception of plot coherence and character believability.

1.4. Reader's Guide

There are many parts to this dissertation. Depending on the interests of the reader, it is not necessary to read the entire dissertation, nor read the entire dissertation sequentially. The following describes what the reader will find in each chapter.

Chapter 2 is a discussion of related work. The chapter summarizes other, relevant research in story generation in order to motivate a planning approach to the problem of story generation. The chapter starts out by presenting a framework for evaluating story generation systems and then describes how the strengths and weaknesses of various classes of story generation systems are applicable to the problem of generating stories with both strong plot coherence and strong character believability. Exemplar story generation systems from each class are described to illustrate those strengths and weaknesses.

Chapter 3 discusses the properties of planning technologies that make a planning approach to story generation one that can solve the problem of generating stories with strong plot coherence and strong character believability. This chapter describes a model of dramatic authoring and describes how a planning approach can be used to approximate the process model. The ways in which planning does not naturally correspond to dramatic authoring are enumerated and I present ways in which models of plot coherence and character believability can be incorporated into the planning algorithm.

Planning algorithms are not perfectly suited to story generation. As general problem-solvers, planners do not necessarily generate plans that have plot coherence or character believability. Chapters 4, 5, and 6 describe extensions to existing planning algorithms (e.g. partial-order planning) that specialize planning to story generation. Chapter 4 describes a planning algorithm that considers the intentions of the story world characters separate from the intentions of the author in order to ensure that characters in the story plan appear to be acting on their own beliefs, desires, and motivations. Intentionality is one component of character believability. Chapter 5 describes a planning algorithm that uses a model of personality to determine whether characters in the story world are acting consistently. Consistency is another component of character believability. Chapter 6 describes a planning algorithm that enables a story generation system to assume responsibility for determining certain characteristics of the story world and the characters in the story world in order to better fit the model of dramatic authoring presented in Chapter 3.

Bringing the three planning algorithms together yields a single narrative generation system, *Fabulist*. Chapter 7 describes how the story planner can be combined with a discourse planner and a media realizer to generate *narrative* – the recounting of the events that were planned by the story planner. The Fabulist system is described as the combination of story planner, discourse planner, and media realizer. The Fabulist architecture is based on the structurist perspective of narrative described in Section 1.2. Chapter 8 describes some of the limitations of the Fabulist system and proposes future work to be done to overcome these limitations. Chapter 9 describes an empirical evaluation of the Fabulist system, focusing on the story planning component. The purpose of the evaluation is to determine whether the advances in planning for story generation significantly increase the audience's perception of plot coherence and character believability in computer-generated stories. Chapter 10 summarizes the research on narrative planning and presents conclusions that can be drawn from my work.

Chapter 2

Related Work

The research problem I am addressing is how to automatically generate stories that are both strong in character believability and strong in plot coherence. Therefore, it is of interest to analyze the degree to which related story generation systems are capable of generating stories with character believability and plot coherence. This chapter presents a framework for evaluating story generation systems that highlights the limitations of related research systems with regard to their reliably generating stories with strong character believability and strong plot coherence. While the limitations of some of the systems that I describe below arise, in whole or in part because of the distinct nature of the research problems that they address, it is an informative exercise to identify the attributes of other systems that are valuable to the problem I am solving and to motivate my solution by highlighting differences. This chapter reviews the story generation systems that are most relevant to my work.

2.1. Evaluating Story Generation Systems

It is difficult to compare the abilities of all previous work on story generation systems. To do so, not only would one need to have access to the systems, but each system would have to be made to generate similar stories to their best capability. Since I do not have access to nor can all systems be made to generate stories that can be directly compared, direct comparison is not possible. It is possible, however, to sort story generation systems into one of a few broad categories based on design philosophy and then to make general observations about each category. Specifically, I am interested in how each category relates to the concepts of character believability and plot coherence.

2.1.1. Overview of Existing Frameworks

Bailey (1999) was the first to categorize story generation systems, dividing systems informally into three categories:

- *Author models*, in which story generation is approached from the perspective of how a human author would invent a story.
- *Story models*, in which story generation proceeds from an abstract representation of the story as a structural or linguistic artifact such as a grammar.
- *World models*, in which story generation is approached by constructing a virtual world and then simulating characters within the world.

Mateas and Sengers (1999) present a scheme (adopted from (Bailey, 1999)) that categorizes story generation systems as *author-centric*, *story-centric*, or *character-centric*. These categories are identical to Bailey's categories except that Bailey calls character-centric systems world models. Sobral, Machado, and Paiva (2003a) distinguish between two categories of systems that use or generate story: those that are *plot-based* and those that are *character-based*. Plot-based systems represent story as discrete, high-level pieces of plot that are interrelated through explicit linkages. Character-based systems involve autonomous characters that can act freely in the story world. This scheme is primarily meant to distinguish design approaches to building interactive narrative systems in which the user is a participant in a virtual world in which a story is being told. However, the scheme can also be applied to story generation systems.

Mateas and Stern (2000) present a continuum between *strong autonomy* and *strong story*. The strong autonomy position states that characters should have the maximum amount of autonomy to make decisions and act in the world. The strong autonomy position advocates the decoupling of character from story. The strong story position, on the other hand, states that a single global, decision-making process should decide what actions every story world character takes at every moment in time. The strong story position advocates the coordination of characters in order to achieve dramatic effects. Strong autonomy is similar to the character-centric categorization and strong story is similar to the author-centric categorization, although the continuum allows for a system to fall somewhere in between the extremes. The continuum however, only addresses the degree to which the decision-making of individual characters is distributed – or apparently distributed – versus centralized.

2.1.2. A Revised Framework for Categorizing Story Generation Systems

Simplifying Mateas and Sengers (1999), I categorize story generation systems as *character-centric* and *author-centric*.

> **Definition 2.1 (Character-centric):** *A character-centric system models the mental factors such as beliefs, desires, goals, and traits that affect how characters act in a virtual world and simulates the cognitive processes of those characters as they react to the environment they are situated in. As time progresses forward in the simulated virtual world, characters respond to the world and story emerges from the decisions that the characters make.*

> **Definition 2.2 (Author-centric):** *An author-centric system computationally models the creative process of a human author. A single authoring agent implicitly or explicitly has intentions towards the outcome of the story or the structure of the plot.*

The story-centric category of Mateas and Sengers is incorporated into the author-centric category. The primary distinction between the categories in Mateas & Sengers (1999) and Sobral, Machado, and Paiva (2003a) is that the latter does not include a category for grammar-based systems. However, one can consider that the *story-centric* category to be a subset of the author-centric category. A grammar can be used to model cognitive structures of memory in a person. Specifically, a grammar can be used to model the way in which humans comprehend of narratives (Thorndyke, 1977). If a grammar such as that by (Rumelhart, 1975) or (Thorndyke, 1977) can be a cognitive model of narrative comprehension, it can also be a model of a process of story generation that is cognitively

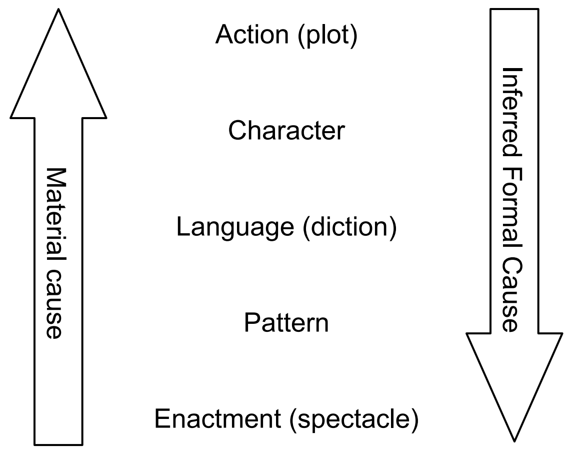

Figure 2.1. Aristotelian theory of drama.

plausible[3] (in the same way that the domain information used to infer understanding from an agent's goal-driven behavior can be used to generate goal-driven behavior (Wilensky, 1981)). Even Hitch (Bailey, 1999), for which the story-centric category was defined, uses a model of reader comprehension of narratives to guide the story generation process. Generalizing this argument, I claim that story-centric systems implement computation models of story authoring, making them a type of author-centric systems.

Laurel's (1991) analysis of Aristotle's *Poetics* describes stories (plays) in terms of the five hierarchical categories represented in Figure 2.1 (Laurel, 1991, fig. 2.2). The categories are related by formal cause and material cause. Formal cause is a top-down perspective that suggests that a story has some purpose and thus dictates the types of characters that can be in the story and the actions that they perform. Material cause is a bottom-up perspective that a layer is made up of the materials at the lower levels. The author views story from the perspective of formal cause. The audience views story from the perspective of material cause because they observe the physical enactment and infer from what they observe character and plot.

From the Aristotelian perspective, character-centric systems generate stories from the perspective of character or lower. Character-centric systems model the cognitive processes

[3] See (Black & Wilensky, 1979) for a critique of story grammars which claims that grammars accept some ill-formed stories and exclude other well-formed stories. The practical consequence is that story grammars may not be suited to story generation.

of the story world characters and simulate their actions in a virtual world. The simulation usually involves forward progression of time so that the implications of character actions can be fully captured and reacted to by other characters. Story emerges from the actions that characters perform in the story world (Aylett, 1999). In these systems, character is the material with which plot is made. Author-centric systems, on the other hand, generate stories from the perspective of the plot. Author-centric systems treat characters as the effectors through which the hypothetical author changes the world (Egri, 1960). Author-centric systems possess some degree of intention towards what the structure of the plot should be or what the outcome of the story should be and use characters to achieve those authorial intentions.

2.1.3. Comparing the Framework to Properties of Story

The framework for categorizing story generation systems is useful for making broad distinctions between various approaches to the problem of generating stories. But what does it mean to use a character-centric approach over an author-centric approach, or vice-versa? The assumption made in Section 1.3 is that successful stories will have both strong character believability and strong plot coherence. However, it is my belief that the general design approach – either character-centric or author-centric – biases a story generation system towards achieving either strong character believability or strong plot coherence. Specifically I make the following general observations.

- Character-centric systems tend to more reliably generate stories with strong character believability but tend to less reliably generate stories with strong plot coherence.
- Author-centric systems tend to more reliably generate stories with strong plot coherence but tend to less reliably generate stories with strong character believability.

These observations are not hard-fast rules. Certainly it is possible for a character-centric system, for example, to generate a story with strong plot coherence. In general, plot coherence is stories generated by character-centric systems is either a fortunate side-effect or accidentally achieved. The converse applies to author-centric systems and character-believability in the stories that they generate. It is possible that character-centric systems exist or can exist that reliably produce stories with both strong character believability and strong plot coherence. Likewise, it is possible that author-centric systems exist or can exist that reliably produce stories with both strong plot coherence and strong character believability.

Whether or not a story generation system is character-centric or author-centric suggests broad implications about its ability to generate stories with character believability and plot coherence. While character-centrism and author-centrism are defined in opposition to each other, character believability and plot coherence, as properties of story, are not opposite ends of a spectrum. That is, a story is not either character believable or plot coherent or somewhere in between. Instead, character believability and plot coherence are two dimensions in the Cartesian space of stories; a story can be both character believable and plot coherent. The assumption given in Section 1.3 is that successful stories are strong in character believability and plot coherence. However, the general observations about character-centric systems and author-centric systems would suggest a generated story can have either strong character believability or strong plot coherence.

2.1.3.1. General Observations about Character-Centric Systems

In general, character-centric systems more reliably generate stories with strong character believability and less reliably generate stories with strong plot coherence. That character-centric systems are proficient at producing stories with strong character believability is evident from the definition of character-centricity. Character-centric systems simulate a virtual world in which characters independently pursue their individual goals in a manner consistent with their traits and desires. Since the actions that characters perform can be observed by the audience to be motivated by the goals, desires, and traits of the character themselves, character believability is a natural side-effect of character-centric systems.

The responsibility for story production is distributed to the characters and story emerges from the goals they adopt, the decisions they make, and the behaviors they perform in the virtual world (Aylett, 1999). Emergent narrative results in what is called the *narrative paradox*

(Aylett, 2000) meaning that allowing autonomous characters (or even an interactive user) authority to make decisions independently of other agents makes plot coherence unlikely. Decision-making at the level of the character is necessarily localized in scope; characters react to the world around them and, if modeled after real world counterparts – humans – are not concerned with the long term consequences of their actions on the perceived development of a plot. Plot coherence is something that is only evident to the audience after long term exposure to material cause.

To compensate for difficulties achieving plot coherence due to the narrative paradox, some character-centric systems utilize additional modules, often called *drama managers* (e.g. Bates, 1992; Kelso, Weyhrauch, & Bates, 1993; Weyhrauch, 1997; Lamstein & Mateas, 2004), to monitor the emergence of story and coerce or constrain characters to behave in ways that are more likely to yield coherence of plot. Unfortunately, any external manipulation of character autonomy necessarily decreases the believability of characters since characters may suddenly adopt new goals, make decisions, or behave in ways that are unexpected, inconsistent, or seemingly lacking in motivation for the sake of plot development.

2.1.3.2. General Observations about Author-Centric Systems

In general, author-centric systems more reliably generate stories with strong plot coherence and less reliably generate stories with strong character believability. Author-centric systems place responsibility on production of story on a single agent – the author – that models the process of creating stories used by some hypothetical human author. Characters are not first-class entities but are instead treated as the author agent's effectors of change in the story world (Egri, 1960). Author-centric systems have intentions towards the outcome of the story or the structure of the plot – the way in which the outcome is achieved. In some instances, the outcome of the story can be the achievement of goals that characters have to begin with. In other instances, the outcome of the story can be declared at the higher level of action or plot (see Figure 2.1). Goals at the plot level – author goals – describe what the world should be like after the story ends without regard to what goals the story world characters might have. The author agent decides what actions story world characters take and the choices that are made are in attempt to achieve author goals or a particular plot which guarantees strong plot coherence.

Character believability often takes a second seat to plot coherence in author-centric systems, however. Author goals and character goals are often not compatible and achieving author goals by choosing the actions that characters take in the story world can cause characters to appear to act inconsistently or without motivation. The author agent can decide what each story world character needs to do and when to do it in order to bring about a particular conclusion. The character actions might not make sense to the audience since the audience will attribute the actions that are performed to characters. The audience assumes that the characters know nothing about the goals of the author and are acting in response to the world they are situated in. Systems that treat the achievement of characters' initial goals or needs as the plot outcome generate stories that are more likely to have character believability since character actions truly are motivated; however, the range of stories that can be told by these systems is limited. Character believability is, consequently, not typically considered by author-centric systems to the degree that plot coherence is.

To compensate for loss of character believability some author-centric systems employ techniques for enhancing the believability of the story world characters by implementing a personality model or by using post-processing to explain why characters might be motivated to act the way they do. A personality model, however, typically only filters unlikely character actions but does not inform the generative process in a way that it can reason about the motivation characters might have for performing actions. Post-processing can only go so far to provide believability since it can only add to the existing structure. If the outcome of a story is the aggregation of all the goals of all story world characters, then all character actions will be believable if those characters only perform actions that are necessary for their individual goal (a subset of the outcome). However, the space of stories that can be told when the outcome is a combination of character goals is limited. That is, the story can only be about how a set of characters cooperate to achieve some world state.

2.2. Review of Story Generation Systems

Research into story generation and the development of story generation systems began as a natural progression from cognitive science research in story understanding in the 1970's. Since then, numerous systems have been developed which generate stories or can be considered to generate stories. Story generation systems include interactive drama systems in which the user can participate in a storytelling experience by entering a virtual world through an animated avatar. Although in many cases, interactive drama systems rely on pre-scripted story fragments, because the user can intervene with other character's goals and with the way in which a story unfolds, an interactive drama system must be able to adapt or restructure the story in order to present a unified experience to the user. The historical record of the events that occur in the virtual world is a story that was generated dynamically.

Given the vast number of systems that can be described as story generation systems, it is beyond the scope of this dissertation to enumerate and describe all of them. The story generation systems described in this section are those that are most relevant to the research on story generation presented in later chapters.

2.2.1. Character-Centric Systems

Character-centric story generation systems are those that model story generation as the simulation of characters in a story world. The character-centric approach draws parallels between art and life. We, as humans, build cognitive structures that represent the real events in order to understand the world around us. Furthermore, there is evidence that suggests that we employ the same cognitive processes to understand narratives (Bruner, 1990). The assumption therefore is that in order for stories to be compelling, story world characters should behave as if they were real people and the story world were the world. In the real world, people are motivated by their individual goals and we understand and predict what others do by inferring their goals. We apply those same processes to the stories we attend to by observing the behaviors of the story world characters, infer their goals, and predict their future actions (Gerrig, 1993). Character-centric systems initialize a story world with characters that have goals and desires. The characters are then simulated by various computational methods so that they act in order to fulfill those goals and desires. Due to the emphasis on character, character-centric systems often, although not necessarily, use autonomous agent technologies such that each story world character is represented by an intelligent agent. Due to the agent-oriented nature, character-based approaches are favored by systems in which the user is allowed to act on story world because characters are represented by autonomous agents that are able to react to unanticipated user interventions (Sobral, Machado, & Paiva, 2003b).

2.2.1.1. Tale-Spin

Tale-Spin (Meehan, 1976) is a system that generates Aesop Fable-like stories by modeling characters in a story world. The user describes one or more characters to Tale-Spin and gives at least one of the characters a goal to achieve. In addition to goals, characters have knowledge about some parts of the story world and have attitudes towards each other. Tale-Spin reasons about how each character, given what it knows about the world, can pursue its goals. The model of reasoning used to simulate the story world characters is based on Schank's model of human knowledge structures (Schank & Abelson, 1977). On behalf of each character in the world, Tale-Spin employs a combination of planning and inference to determine the next action that character should take (Meehan, 1976). The character may have a pre-existing script which can be executed or may have one or more partial plans that can be completed. A character tries each option in turn until it achieves its goal or fails. After each step of a script or a plan is executed, an inference engine determines how the story world has changed and how the characters in the story world react to the change. The inference process can result in characters adopting new goals.

Since characters are driven by goals and desires and their reasoning is modeled on a theory of human reasoning, the stories generated by Tale-Spin have character believability. The audience is able to perceive that the behaviors performed by the characters are in pursuit of their goals. Character behavior is planful and goal-based so Tale-Spin stories are also coherent. Tale-Spin, however, does not contain any formal notion of plot beyond the fact that characters have goals that are pursued. If the protagonist is given an initial goal, one could make the case that the successful achievement of the protagonist's goal is a meaningful outcome of the story. In that case, any story in which the protagonist achieves his goal can be plot coherent. There are two reasons that Tale-Spin is not guaranteed to generate stories with plot coherence. First, the system can infer new goals depending on how the world evolves; characters can start pursuing new goals that are not related to the "outcome" of the story. Figure 2.2 shows a story generated by Tale-Spin (Meehan, 1977, p. 91). The two characters were each given the initial problem that they were thirsty but the resulting story appears to be about how George the ant becomes indebted to Wilma the bird. If the rescue is the point of the story, then the part of the story in which Wilma the bird satisfies her thirst is not meaningful to the outcome.

The second way in which Tale-Spin can fail to generate plot coherence is due to the fact that scripts and plans can fail when the story world is not set up in a manner that is conducive for the characters to solve for their goals. If a character is hungry and has a goal to satisfy his hunger but does not know where any food is and does not have any plan to find food (e.g. ask someone who is friendly to him where some food is), then the resulting story will be brief and without plot coherence. Figure 2.3 (Meehan, 1976, p. 95), translate to natural language by Meehan, is a story generated by Tale-Spin that fails because the initial description of the world was not adequate. The story in Figure 2.3 lacks plot coherence because the events in the story do not appear to lead to any meaningful outcome. The inference engine determined that the crow would become hungry when noticing it had some cheese. Had the world been

```
ONCE UPON A TIME GEORGE ANT LIVED NEAR A PATCH OF GROUND. THERE WAS A
NEST IN AN ASH TREE. WILMA BIRD LIVED IN THE NEST.  THERE WAS SOME WATER
IN THE RIVER. WILMA KNEW THAT THE WATER WAS IN THE RIVER. GEORGE KNEW
THAT THE WATER WAS IN THE RIVER. ONE DAY WILMA WAS VERY THIRSTY. WILMA
WANTED TO GET NEAR SOME WATER. WILMA FLEW FROM HER NEST ACROSS A MEADOW
THROUGH A VALLEY TO THE RIVER. WILMA DRANK THE WATER. WILMA WASN'T
THIRSTY.

GEORGE WAS VERY THIRSTY. GEORGE WANTED TO GET NEAR SOME WATER. GEORGE
WALKED FROM HIS PATCH OF GROUND ACROSS THE MEADOW THROUGH THE VALLEY TO A
RIVER BANK. GEORGE FELL INTO THE WATER. GEORGE WANTED TO GET NEAR THE
VALLEY. GEORGE COULDN'T GET NEAR THE VALLEY. GEORGE WANTED TO GET NEAR
THE MEADOW. GEORGE COULDN'T GET NEAR THE MEADOW. WILMA WANTED GEORGE TO
GET NEAR THE MEADOW. WILMA WANTED TO GET NEAR GEORGE. WILMA GRABBED
GEORGE WITH HER CLAW. WILMA TOOK GEORGE FROM THE RIVER THROUGH THE VALLEY
TO THE MEADOW. GEORGE WAS DEVOTED TO WILMA. GEORGE OWED EVERYTHING TO
WILMA. WILMA LET GO OF GEORGE. GEORGE FELL TO THE MEADOW. THE END.
```

Figure 2.2. A story generated by Tale-Spin.

```
ONCE UPON A TIME THERE WAS A DISHONEST FOX AND A VAIN CROW.  ONE DAY THE
CROW WAS SITTING IN HIS TREE, HOLDING A PIECE OF CHEESE IN HIS MOUTH.  HE
NOTICES THAT HE WAS HOLDING THE PIECE OF CHEESE.  HE BECAME HUNGRY, AND
SWALLOWED THE CHEESE.  THE FOX WALKED OVER TO THE CROW.  THE END.
```

Figure 2.3. A "mis-spun tale" generated by Tale-Spin.

instantiated so that the crow had recently eaten, then the crow would not have eaten the cheese and the fox would proceed to trick the crow into dropping the cheese (Meehan, 1976). The protagonist is the fox and his goal to satisfy his hunger. However, if one makes the case that if the meaningful outcome of the story is the satisfaction of the crow's hunger, then the story in Figure 2.3 would be plot coherent if the fox's actions were completely omitted.

2.2.1.2. The Oz Project

The Oz Project (Bates, 1992; Bates, 1994; Mateas, 1997; Loyall, 1997) uses reactive, animated, autonomous agents that interact with human users to create interactive dramas. Agents are implemented in a reactive planning language called HAP (Loyall, 1997) such that each agent has its own set of unique expressible behaviors. As the user, embodied in an animated avatar, interacts with the autonomous agents in a virtual world, the characters form goals and perform behaviors that achieve those goals. Goals are decomposed into a hierarchy of sub-goals that eventually decompose in primitive behaviors that are executed as animated behaviors. Behaviors have applicability conditions that must be true in the world for a behavior to be performed; if there are no applicable behaviors then a goal fails. Expressiveness of an agent is captured in the unique way in which goals are decomposed into primitive behaviors.

Character believability is apparent in the Oz project because each character forms goals and reactively pursues those goals with distinctive personality as implemented by individualized HAP behaviors. However, left to their own means, the virtual world will be unstructured and the autonomous agents will react and interact without direction. Since the agents are situated in the virtual world they are unable to reason about the long-term consequences of their goals and behaviors in terms of whether events are leading towards a meaningful conclusion or not. In order to provide direction to the characters in the world, a module called the drama

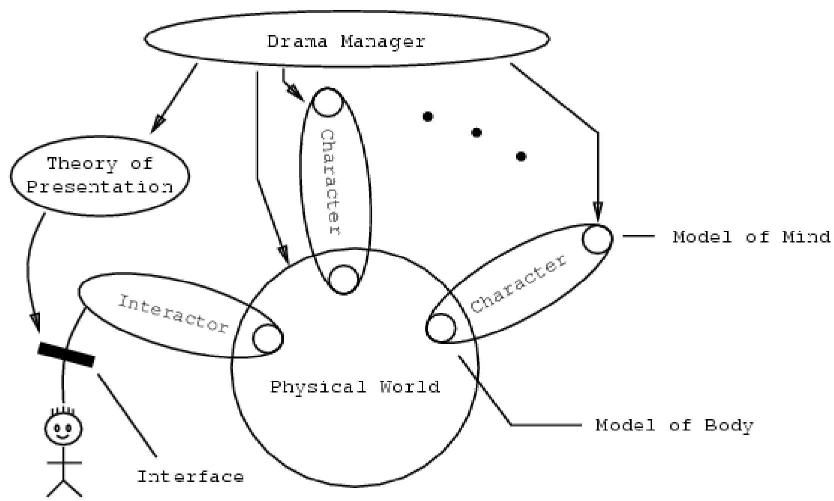

Figure 2.4. The Oz Architecture.

manager is introduced to the architecture. Figure 2.4 (Kelso, Weyhrauch, & Bates, 1993, fig. 1) shows the Oz architecture. The drama manager models plot as a directed, acyclical graph (DAG) of scenes as in Figure 2.5 (Kelso, Weyhrauch, & Bates, 1993, fig. 2). Each scene is an abstract representation of the events that should occur together during a localized block of time in the story world. The drama manager observes the action in the virtual world and advances the story by advancing a frontier. When the drama manager sees an opportunity to advance the story to the next scene it subtly influences the goals and behaviors of the agents in the virtual world so that the scene occurs. Influencing the goals and behaviors of agents is detrimental to character believability because character actions can appear unmotivated when intervention by the drama manager occurs. However, preliminary experiments demonstrate that an interactive user situated in the virtual world is less likely to notice inconsistencies in character behavior (Kelso, Weyhrauch, & Bates, 1993).

The plot graph is provided by the system developer as a way of constraining the possible stories that can emerge from the system. However, since the plot graph provides an abstract specification of the plot, there is room for improvisation in the story world so that each performance of the same plot structure can result in distinct stories as the level of primitive behaviors. The drama manager cannot guarantee plot coherence, however, because agents responding to the world and the user can develop and pursue new goals that are not meaningful to any of the outcomes described in the plot graph. Such sequences of actions

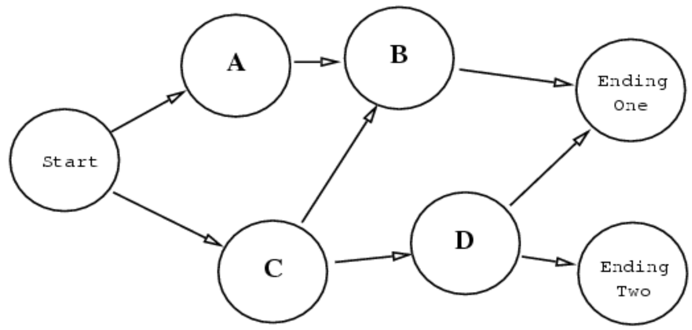

Figure 2.5. A plot diagram in Oz.

can be distinct enough from the scenes in the plot graph that the drama manager is unable to recognize opportunities to coerce the agents back into the prescribed plot. One reason that this occurs in that the user is an interactive part of the story world and can choose to pursue activities that are not strictly part of the plot outline, causing the other agents in the world to respond accordingly.

2.2.1.3. The Virtual Storyteller

The Virtual Storyteller (Theune et al., 2003) is based on the Oz project but does not involve an interactive user. Characters are represented by autonomous agents who pursue their own goals. A drama manager is used to subtly influence the way in which a plot unfolds. The drama manager in the Virtual Storyteller, however, uses a proscriptive method of influencing the agents instead of a prescriptive method. Agents ask permission from the drama manager to perform behaviors in the virtual world and the drama manager can grant or deny a request based on whether that behavior violates certain rules about how plots is expected to unfold. The Virtual Storyteller distinguishes between fabula and sjužet in that the system uses a two phase process, first generating a sequence of events that occur in the story world and then reasoning about how best to relate the fabula to the audience. The fabula is formed by the actions of the agents in the virtual world. The agents are not visually animated as in Oz. Instead, the fabula is narrated by another agent who relates the occurrences of the virtual world in a descriptive, verbal form.

2.2.1.4. The Virtual Theater Project

The Virtual Theater Project (Hayes-Roth & van Gent, 1997) implements autonomous agents in a form of storytelling referred to as directed improvisation. Animated agents create performances in real time without detailed planning. Agents infer what is happening around them in the virtual world and reactively select and schedule relevant behaviors to execute. Agents do not have goals but merely react to the environment and to the other agents in the environment. Instead agents obey the rules of performance improvisation, such as "accept all offers," recommended by Johnstone (1981). To generate story, the agents are provided a scenario that specifies character roles, basic function of the plot, and timing of entrances and exits onto a virtual stage. The scenario is implemented as a set of constraints on agent behavior but the way in which the plot develops within the bounds of the constraints is improvised by the autonomous agents (Hayes-Roth & van Gent, 1997; Hayes-Roth, van Gent, & Huber, 1997). The degree of plot coherence in an improvised story depends on the scenario constraints imposed on the agents by the user. The more constrained the performance of the agents, the less likely that the improvised actions of the agents will be irrelevant to the outcome. Character believability is also a function of the scenario since the constraints can cause agents to change role and interpersonal relationships.

2.2.1.5. I-Storytelling

The Interactive Storytelling, or I-Storytelling, project (Cavazza, Charles, & Mead, 2001) uses autonomous agents to generate stories. The autonomous agents are given goals and use hierarchical task networks (HTN) to generate plans to achieve those goals. The task networks are hierarchies of sub-goals and actions. Each character in the virtual world generates a plan by searching through the AND/OR graph corresponding to the HTN, allowing the agent to interleave planning and execution. The way in which the virtual world is initialized, especially the initial goals that are given to the characters and the initial positions that the characters start at, affect the outcome of the story. If the character goals force the characters to rely on the same, limited resources, then race conditions can occur. The characters that do not get resources are forced to re-plan, affecting the overall emergence of plot. For example, Figure 2.6 (Cavazza, Charles, & Mead, 2001, fig. 3) shows one

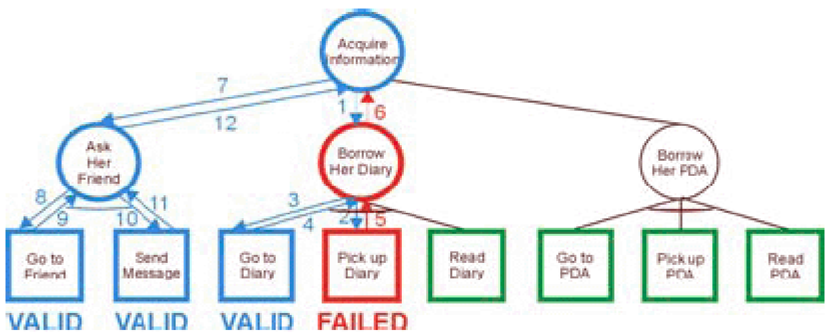

Figure 2.6. A hierarchical task network for a character in the I-Storytelling project.

character's HTN to acquire information about another character. There are three alternative methods for achieving this goal: ask a friend, borrow her diary, or borrow her PDA. The character initially plans to borrow her diary but during execution of this plan, the diary becomes inaccessible. The alternative plan of asking a friend is tried next and succeeds.

As with other story generation systems that use reactive agents, the way in which an agent pursues its goals and whether it succeeds or not affects the way in which plot emerges. Plot coherence cannot be guaranteed because the way in which the story ends cannot be determined. Unlike other character-centric systems, characters are given initial goals and do not adopt new top-level goals during plan execution. Thus, with a sufficiently rich HTN it is possible for all characters to eventually find a successful plan and achieve their goals, regardless of failures due to competition over environment resources. In this case, the emergent story will be plot coherent. The variability of the emergent plotlines and the ability to recover from failure is dependent on the ability of the human author to design a sufficiently detailed and fault-tolerant HTN, at least at the higher levels where goals are fixed. A state-based planning technology such as heuristic search planning provides characters with more flexibility and ingenuity when achieving their goals and enables them to better able to recover from failure (Charles et al., 2003). Regardless of the planning algorithm, character believability is guaranteed because agents are motivated by goals and plan their own actions in the story world.

2.2.1.6. America's Army: Soldiers

America's Army is a research and development initiative consisting of two computer systems: Operations and Soldiers (Zyda et al., 2003). America's Army: Operations is a first-person shooter video game design to train soldiers in combat conditions. America's Army: Soldiers is an interactive narrative system that enables a recruit to enter a virtual world as a character pursuing a career in the U.S. military. The army is presented through a series of sensible, interactive, non-repeating stories. The story engine used in America's Army: Soldiers uses autonomous agents that represent story world characters (Osborn, 2002). The human author that engineers the story world domain exerts a high level of control over the stories that develop in the story world through the use of a drama manager. Agents embodied in the virtual world have their own goals and personalities. Their behaviors are governed by rules of character-to-character and character-to-environment interactions. The goals of the agents motivate the occurrences in the story world. However, to foster more goal-directed stories, HTN-like plans called *tickets* are activated from time to time. A ticket is a general plan for how a scene should play out including how character in particular roles should interact. At any given time, there are many opportunities for an agent to respond to the environment, to respond to other agents, or to become involved in a ticket. An agent uses weighted preferences to determine which interaction to engage in at each moment in time.

The characters in the story world simulated by the story engine are represented by autonomous agents that react to the local environment in which they are situated. Since the agents have goals and act in the world to achieve the goals, stories that emerge from the story engine can be character believable. However, a plot coherent narrative is not likely to emerge from the autonomous local reactions of the characters. Tickets help to direct the emerging storyline towards plot coherence. Tickets are provided by the human author at initialization. The degree to which agent interactions are structured and meaningful to a particular outcome depends on the way in which tickets are authored. Regardless of how tickets are authored, agents can choose not to participate in the plan-like story fragments if their rules of preference lead do not allow them to assume roles in the ticket specifications.

2.2.2. Author-Centric Systems

Author-centric story generation systems are those that computationally model the creative process of an author as opposed to simulating a virtual world. An agent – the author – reasons about the story world and the characters that exist in it to construct a coherent sequence of events that can be told to an audience. One of the primary distinctions between the author-centric approach and the character-centric approach is that author-centric systems are driven by the intentions of the author instead of the intentions of the story world characters. In character-centric approaches, characters intend to achieve certain goals that arise from being situated in the world in a way that is favorable to their well-being. In contrast, the author intends the story – the amalgamation of all the events that happen in the story world – to be satisfying or meaningful to the audience (Dehn, 1981). Story world characters, as part of the story world, do not have the global perception of an author and cannot know whether their actions are going to combine in a way that is dramatic or satisfying to an audience.

Dehn (1981) claims that the process of story generation must be a process that includes the satisfaction of the intentions of the author. The simulation approach leads to stories in which characters act on their goals and desires. How the characters achieve their goals (or how characters fail to achieve their goals) may make for an interesting story. However, there is no guarantee that the way characters achieve their goals will result in interesting interactions between characters or in dramatic situations such as suspense or situation reversal. The way in which the story world is constructed may constrain the characters such that they act in ways that result in a satisfying story, although to set up the world ahead of time is "tricky" (Dehn, 1981). The story can be coherent in the sense that one event follows naturally from prior events but not meaningful beyond the intentions of the characters; the story is not guaranteed to be *plot coherent*. Author-centric systems achieve plot coherence by reasoning from the perspective of an author, therefore considering the author's intentions and reasoning about how to coherently achieve those intentions.

2.2.2.1. Universe

The Universe system (Lebowitz, 1984; Lebowitz, 1985) uses a planning approach to generate open-ended stories. Open-ended stories are those that have no well-defined conclusion and continue episodically, such as soap operas. The planner creates a plan to achieve the author's goals – what the human author would like to see happen during a particular episode – by piecing together plot fragments that achieve the author goals and any sub-goals that are introduced during the planning process. Plot fragments, which are similar to abstract operators used in NOAH (Sacerdoti, 1977) describe situations that occur in the story world at various levels of abstraction. Abstract plot fragments are decomposed by posting sub-goals until a primitive level is reached where the story can actually be told (Lebowitz, 1985). Plot fragments have constraints that must be met for a plot fragment to be applicable as part of a story plan.

The important distinction between Universe and Tale-Spin (Meehan, 1976) is that Universe is driven by the intentions of the author while Tale-Spin is driven by the intentions of the story world characters. The goal of the planning problem describes a situation that the author wants to have happen in the story world. As the planner incorporates plot fragments into the story plan, sub-goals are introduced and subsequently planned for. The planner only reasons about the ways to satisfy the top-level goal and subsequent sub-goals that are all specified by the author as constraints on the way in which the story unfolds. The planner never considers whether or not it makes sense for a character to become involved in a particular plot fragment. Plot coherence is assured because the actions of the story world characters always achieve the author's goal. Character believability however, cannot be guaranteed because character actions are not motivated by anything other than the author's goal.

Universe defines *person frames* for each character in the story world that store particular information about a character, including name, status, role, and traits (Lebowitz, 1984). Whether or not a character can be involved in a plot fragment is determined by whether or not the values in that character's person frame match with the constraints of the plot fragment. For a character to match the role of a plot fragment is sufficient motivation for that character to be involved in a plot fragment (Lebowitz, 1985). No consideration is given by the Universe planner for ensuring that the character appears to be achieving a goal other than the author's goal. For example, `Forced-Marriage` is a plot fragment in which an evil parent forces the protagonist to forsake her lover and marry the evil parent's son. The schema for `Forced-Marriage` as represented in Universe is shown in Table 2.1 (Lebowitz, 1985, table 2). By using the `Forced-Marriage` plot fragment in the story plan, the planner achieves the goal of preventing two characters – a protagonist and her lover – from being happy. The evil parent threatens the protagonist in order to force her to marry his son. The schema, however, does not address why the evil parent wants his son to marry the protagonist except that it prevents the protagonist from being happy. From the perspective of the audience, the story involving the forced marriage is coherent – the audience can follow the sequence of events that results in the marriage – but not believable. By observing the actions of the characters, the audience will not be able to understand the motivations of the evil parent.

2.2.2.2. Tailor

Tailor (Smith & Witten, 1991) is an author-centric story generation system that uses search to plan the actions of the story's protagonist. The protagonist is given a goal to achieve and Tailor searches for a sequence of actions the protagonist can take to achieve the goal. In order to introduce conflict into a story, Tailor allows a second character – an antagonist – to be part of the story. The antagonist does not have a goal except to ensure that the protagonist

Table 2.1. A typical Universe plot fragment.

```
PLOT FRAGMENT: Forced-Marriage
CHARACTERS: ?him ?her ?husband ?parent
CONSTRAINTS: (has-husband ?her)      {the husband character}
             (has-parent ?husband)   {the parent character}
             (< (trait-value ?parent 'niceness) -5)
             (female-adult ?her)
             (male-adult ?him)
GOALS: (churn ?him ?her)
SUBGOALS: (do-threaten ?parent ?her "forget it") {threaten her}
          (dump-lover ?her ?him)     {have ?her dump ?him}
          (worry-about ?him)         {have someone worry about ?him}
          (together * ?him)          {get ?him involved with someone else}
          (eliminate ?parent)        {get rid of ?parent}
          (do-divorce ?husband ?her) {end the unhappy marriage}
          (or (churn ?him ?her)      {either keep churning or}
              (together ?her ?him))  {try and get ?her and ?him together}
```

does not achieve his goal. When an antagonist is present, Tailor uses an adversarial search. The protagonist and antagonist alternate taking turns to act. The protagonist chooses actions that maximize his chance of achieving his goal and the antagonist chooses actions that minimize the protagonist's chance of achieving his goal.

The stories generated by Tailor are coherent because the search process ensures that one action leads to the next. If the search process were controlled so that the protagonist always manages to achieve his goal despite the actions of the antagonist, then the stories would also be plot coherent. Tailor does not control the search process in this way, however. As with many author-centric systems, character believability is problematic. The protagonist can be believable because his actions are driven by his goal. The antagonist, however, is not necessarily believable because he can take any action that is contrary to the protagonist's goal. Causing another character to fail can be considered a goal, but it is never made clear that that is what the antagonist is trying to achieve and the reason why the antagonist wants the protagonist to fail is never motivated.

2.2.2.3. Mexica

Mexica (Pérez y Pérez & Sharples, 2001) is an author-centric system that models the process of story generation as the process of creative writing. The model of creative writing used in Mexica (Sharples, 1999) conceptualizes writing as a cycle of cognitive engagement and reflection. During engagement, the author probes her long-term memory for new ideas. During reflection, existing ideas are refined and organized. During the engagement phase of processing, Mexica uses a case-based approach in which it probes a database of known existing stories for elements that match current patterns of emotion and tension in the story being generated. The story elements are actions that are chosen to complete sequences because they were used successfully in existing stories. Engagement continues until the system reaches an impasse that cannot be rectified by probing existing stories and switches over to the reflection phase of processing. During this phase, Mexica uses a partial-order planning technique to satisfy the preconditions of actions inserted into the story from the engagement phase. Reflection continues until the story is sound (i.e. for any one event in the story, the sequence of events that precede it is sufficient to establish a world state in which that event can occur) and then returns to engagement.

The reflection process ensures that stories generated by Mexica are coherent because every action will have its preconditions reasonably satisfied by prior actions. However, the plot coherence and character believability of a story relies on the content of the case base of existing stories. If the existing stories that a newly generated story is modeled after are plot coherent, then the new story may also be plot coherent. There is no guarantee that the engagement process will select a coherent sequence of actions. That is, one action does not necessarily have to rely on any previous actions. Likewise, character believability is only accidentally achieved if the engagement process selects actions that explain why characters appear to have the goals that they do. Mexica overcomes the limitation to character believability with an additional process called *final analysis* that operates on the completed story and attempts to add explanations that make character behavior understandable to the audience (Pérez y Pérez & Sharples, 2001).

2.2.2.4. Façade

Façade (Mateas & Stern, 2002; Mateas, 2002) is a system designed to address the difficulties encountered in the Oz Project when adapting autonomous agents to a storytelling system. Specifically, autonomous characters may form goals that are not relevant to the unfolding of the story and perform actions to achieve them. A drama manager that infrequently intervenes to ensure that scenes are carried out is constantly in conflict with the agents (Mateas & Stern, 2002; Mateas, 2002). Instead of relying on autonomous agents and a drama manager that works on the level of the scene, Façade explicitly addresses the balance of character and plot by implementing a reactive behavior planner that selects, orders, and executes fine-grain plot elements called *beats*. A beat is the smallest unit of story structure that can move the story forward (e.g. a bit of dialogue or an action). Frequently, a beat involves an action/reaction pair where one character performs an action and another character reacts accordingly. Such joint behaviors allow fine-grained coordination between characters.

Beats are implemented in a reactive behavior language, ABL (Mateas & Stern, 2002; Mateas, 2002), based on HAP (Loyall, 1997). The most significant contribution to ABL is the

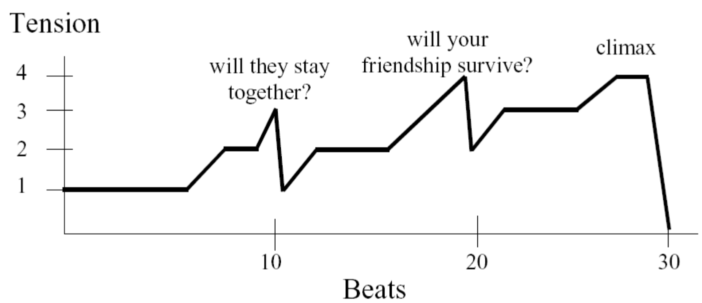

Figure 2.7. The Façade dramatic arc.

extension of HAP goals to support the coordination of multiple characters. Joint goals are satisfied by joint behaviors which describe how more than one character acts in parallel or in sequence. Character behaviors are coordinated without each character having to monitor the other's progress. Joint goals are not owned by any one character, enabling characters to achieve goals that are related to plot progression and not related to an individual character's needs or desires. Beats are selected and sequenced by a type of drama manager called a beat manager. The beat manager identifies beats that are applicable in the current state of the world and sequences the one that is most likely to achieve a particular story arc. A story arc is a function such as that in Figure 2.7 (Mateas, 2002, fig. 8-4) that maps time (as measured by the passing of beats) to the tension that the audience should experience.

Character believability is apparent in Façade because of the rich repertoire of character behaviors and the way that character traits are scripted into the beats. The system uses a set of beats that encode individual character goals and joint story goals so that the characters appear to be motivated by their own goals, desires, and traits while the system simultaneously pursues a well-defined story arc. Plot coherence is also apparent in Façade because beats are selected that conform to the story arc as a function of time and audience tension. By choosing beats that increase tension, Façade is ensured to select beats in an order that develops the plot. There are only a finite number of beats and thus a limited number of conclusions. The user's involvement as a character in the story provides variation in the order in which beats are sequenced but the beats are scripted to ensure that the story never deviates too far.

The primary limitation of Façade is the dependence of the system on the human authoring of beats (Mateas & Stern, 2002; Mateas, 2002). Character believability is only accomplished if the beats are authored such that characters appear to be motivated by individual goals and traits. For example, the entire story can be achieved through joint goal beats that have relevance towards the outcome intended by the author but do not demonstrate any motivation for the goals the characters appear to be pursuing. Consequently, the story will likely not be perceived by an audience as having character believability. Plot coherence too is dependent on the human authoring of beats. The beats must be limited to the scope of behaviors that are appropriate for the story so that, no matter what happens, the next beat that is sequenced moves the story towards its conclusion.

2.2.3. Relationship between Fabulist and Previous Work

The goal of my research is to build a story generation system that generates stories that have both strong plot coherence and strong character believability. Character-centric systems tend to reliably generate stories with strong character believability but less reliably generate stories with strong plot coherence. This is due to the emphasis placed by these systems on autonomy of character and the nature of a simulation approach. An autonomous character does not have the ability to perceive its own actions within the global scope of emergent story. Author-centric systems tend to reliably generate stories with strong plot coherence but less reliably generate stories with strong character believability. This is due to an emphasis on plot structure which is often achieved by forcing characters to perform actions that bring about structured plot instead of performing actions that appear motivated. To generate stories reliably with both strong plot coherence and strong character believability, I adopt an approach that borrows elements of both character-centric and author-centric approaches.

Fabulist is an author-centric story generation system that incorporates character-centric techniques. Specifically, Fabulist uses a partial-order planning approach that has been modified to reason about both author intentions and character intentions. As a system that uses planning, Fabulist is most closely related to the Universe system (Lebowitz, 1984; Lebowitz, 1985). Universe achieves strong plot coherence by planning to achieve authorial intentions. Author goals differ from character goals in that they describe how a plot fragment ends instead of describing something that a character wants to achieve. The drawback, of course, is that a character performs actions that bring about the author goals instead of performing actions that make sense for the character in a given situation. There is no mechanism that reasons about the believability of character actions; if a story fragment has strong character believability, it is entirely accidental. Universe makes some attempt to ensure that character act consistently by enforcing a character trait model. However acting consistently with one's traits is not equivalent to being motivated to act. For example, if a character has a niceness value of -10, it is consistent for that character to force someone to marry his son, but it does not mean that the character has a reason to force someone to marry his son.

Fabulist is also driven by author goals. In Fabulist, a human author specifies what the outcome of the story should be and Fabulist finds a sequence of character actions that result in the specified outcome. However, Fabulist also reasons about character intentions. That is, for any character action that might be part of the story, Fabulist determines whether that action appears believable. If the action does not appear believable, Fabulist repairs the story plan by finding a reason why the character would want to perform that action or by backtracking and trying another character action. Characters in the resulting story appear to form intentions and act to achieve those intentions. As a consequence of the way the characters act, the author goals are achieved. Universe, although it operates on the principles of planning, does not implement backtracking and instead relies on heuristics to pick the best alternative for every decision point (Lebowitz, 1987).

2.3. Summary

The purpose of this chapter is to motivate the use of planning as a technique for story generation. In this chapter, I present a framework for evaluating story generation systems that categorizes systems as character-centric or author-centric. Character-centric systems model the beliefs, desires, goals, and traits of characters in the story world and simulate the characters in a virtual world as autonomous agents. Story emerges from the behaviors performed by characters in the virtual world. Author-centric systems model the cognitive processes of a hypothetical human author who constructs a story as an artifact. I make the claim that character-centric systems more reliably generate stories with strong character believability and less reliably generate stories with plot coherence. The evidence for this claim is that character-centric systems place emphasis on the autonomous decision-making processes of the characters as if they were complete agents. Characters are more likely to react to the world they are situated in and demonstrate intentional behavior. I also make the claim that author-centric systems more reliably generate stories with strong plot coherence and less reliably generate stories with character believability. The evidence for this claim is that author-centric systems view story structure from a global perspective and are more readily able to reason about causality and temporality. However the emphasis on causality and temporality, possibly in the context of achieving some story goal posed by the human author, often takes precedence over reasoning about what the autonomous goals of the characters might be.

This chapter also summarizes relevant story generation systems. Following my framework for evaluating story generation systems, the discussion of related research projects is broken up into descriptions of character-centric and author-centric systems. The relationship between previous work and the approach used in Fabulist is discussed. Since Fabulist uses a planning approach, emphasis is placed on comparing other author-centric planning approaches to Fabulist. Since Fabulist is a planning system, it attempts to achieve a goal – a partial description of the state of the world after the story ends. However, unlike other planning approaches, Fabulist also performs reasoning about the intentions that characters in the story world might have in order to establish character believability by building additional story structure that motivates why those characters commit to their goals.

Chapter 3

Planning and Story Generation

In the previous chapter, I began to draw distinctions between the approach to story generation used in the Fabulist system and other story generation systems. Fabulist is an author-centric system that uses planning to generate stories. I have criticized planning systems such as Universe (Lebowitz, 1985) for not placing enough emphasis on character believability. Although Fabulist also uses a planning approach, Fabulist takes steps to improve the believability of story world characters. The purpose of this chapter is to present rationale for using planning to generate stories. Narratologists distinguish between the fabula, sjužet, and medium of a narrative (Bal, 1997). Each level provides a way of describing or analyzing a narrative. The fabula of a narrative, being a chronological ordering of the events that change the story world, is a representation of the story itself (Prince, 1987). The sjužet and medium of a narrative concern the telling of the fabula. Ignoring, for a moment, considerations about story generation involving discourse, plans, as data structures, share many similarities with fabulas. Search-based planning algorithms, likewise, have many parallels with the process of authoring itself. Although there are some limitations to using planning for story generation – one of which being a disregard for character believability – the similarities warrant a closer inspection of planning as a way of generating a story fabula. In addition to presenting the rationale for using search-based planning algorithms to generate stories, this chapter also lays the groundwork for addressing the limitations of planning as a story generation technology.

3.1. A Computational Model of Dramatic Authoring

The framework for evaluating story generation systems in Section 2.1 focuses on two properties of story: plot coherence and character believability. The framework suggests that story generation systems are either biased towards plot or towards character. One conclusion that one might draw is that plot and character are in some way related. In fact, this is so. Character is the material cause of plot (Laurel, 1991). That is, the decisions that characters make and the actions that they perform in the story world make up the plot. Plot, however, can also be considered the formal cause of character. That is, the plot is the story (or at least an abstraction of the story) and thus constrains what characters can be in the story and what those characters can do. The inseparable nature of plot and character is due to the fact that the audience learns about both plot and character over time through observations of enactment. Since character is at a lower level in the hierarchy (Figure 2.1), the audience necessarily learns about the characters first. The audience first sees the plot of the story as the inevitable result of the characters that exist in the story world; the characters are the material cause of the plot. However, once the audience learns the plot (this may only be at the very end of the story), it is possible for the audience to analyze the story in terms of plot, in which case it is evident that the characters must be the way they are or there would have been no story (Laurel, 1991).

A consequence of the inseparability of plot and character is that if one were to change then the other must necessarily change. To change a character, the plot necessarily changes. Changing anything about a character, such as its beliefs, desires, or traits, means changing the way in which that character reacts to the world and makes decisions. If a character makes even one decision differently than before, then the character will act to change the world in a different way. Since plot emerges from character actions, the plot is necessarily different once the modified character makes one decision differently. To change the plot of a story, characters necessarily change. Changing the plot of a story means that a different sequence of events occurs in the story world. Since the events in the story world occur because of character actions, the characters in the story world must necessarily make different choices and therefore express different beliefs, desires, or traits.

To generate a story that reliably generates stories with both strong plot coherence and strong character believability, the story generation system cannot consider plot distinct from character and vice-versa. A computational model of the authoring process that considers both plot and character is necessary if a new story generation system is to avoid the pitfalls of previous systems. The following sections lay out such a computational model of the authoring process.

3.1.1. A Model of Dramatic Authoring

Examples of computational models of story generation include

- Models of the organization of memory and thought (e.g. story grammars such as (Rumelhart, 1975) and (Thorndyke, 1977)),
- Models of problem solving (e.g. planning algorithms used for authorial planning such as those used by (Lebowitz, 1985) and (Smith & Witten, 1991)),
- Models of rational agent behavior (e.g. deliberative planning as used by (Meehan, 1976) and reactive planning as used by (Loyall, 1997)), and
- Models of creative writing (e.g. the model used in (Pérez y Pérez & Sharples, 2001)).

The models above, as discussed earlier, bias the process towards either plot or character. Aristotle (350 B.C.E) claims that plot should be the first consideration of any author while character holds second place. However, most scholars of modern drama have reversed the standpoint and consider character the most important aspect and plot secondary (Egri, 1960; McKee, 1997).

The model of the process of story generation I have adopted is a model of dramatic writing suggested by Lajos Egri (1960). Egri describes character as the material out of which stories are created. The characters that inhabit the story world are envisioned by the author in their full complexity and are then "set free" to act in the story world as if they were real people in

1. Define a premise.
2. Define a set of characters in complete detail.
3. Simulate the characters in the story world from the beginning. As soon as it is evident that the premise will not be demonstrated, go to step 2.

Figure 3.1. Egri's model of the dramatic authoring process.

the real world. Whatever plot emerges is a result of the decisions and the actions that the characters perform. There must, however, be a guiding force behind the process or the characters, no matter how well formed or how interesting, can take the story in a meaningless direction. Egri proposes the *premise* as such a guiding force. The premise is what the story is about. Premise can mean many things, including theme, thesis, root idea, central idea, goal, aim, driving force, subject, purpose, plan, plot, or basic emotion. An example of a premise is the propositional statement, "Frugality leads to waste." If one were to treat a story as an argument, then the premise is something that must be demonstrated, or "proven." That is, the actions of the characters acting freely in the story world should convince the audience that the premise is true. If the characters do not appear to be taking the plot in a direction that proves the premise without some sort of outside influence exerted by the author, then the author should revise the characters or select new characters. As argued before, the characters of a story cannot be changed without changing the plot of a story. The author must start again from the very beginning of the story and see if the new set of characters can prove the premise. The process of revision and simulation continue until a plot is created that proves the premise. Egri's model of dramatic authoring process is summarized in Figure 3.1.

A premise has three parts: character, conflict, and conclusion. A character is not complete without the environment in which the character belongs (Egri, 1960); the process of defining character is analogous to defining the initial state of the world in which the characters exist. Conflict is an action and the counter-action that it evokes. Conclusion is the way in which the story ends: the final state of the story world. Therefore, a premise – the guiding force behind a story – is simply the initial state of the story world, an action/reaction pair that describes the primary conflict, and the final state of the story world. For example, the premise, "frugality leads to waste," can be captured as a frugal character that refuses to pay taxes, is subsequently penalized by the state, and ends up poor (Egri, 1960).

3.1.2. Computational Formalization of the Model of Dramatic Authoring

Egri's model of the dramatic authoring process is character-driven. At first glance, the model would appear to suggest a character-centric approach since character-centric approaches are simulations of characters in the story world. Character-centric story generation systems define a set of characters with individual beliefs, desires, and traits and then allow them to act autonomously to form and achieve goals. However, there is one aspect of Egri's model that character-centric systems fail to address: backtracking. Given too much freedom, a character-driven approach is not guaranteed to generate a coherent plot. To increase the likelihood of a coherent plot, the premise can be computationally modeled as a drama manager. Egri, however, disapproves of any extra-diegetic intervention[4] on the decision-making of the story world characters. For a character-centric system to be a computational formalization of Egri's model of the dramatic authoring process, the system would have to be restarted over and over until an appropriate set of characters – given some computational formalization of character traits and goals – was found by a human author. Such a process is analogous to a search through the space of possible characters. The search is complete when a set of characters is found that when set free in the story world necessarily interact in a way that demonstrates the premise.

A computational formulation of the model of dramatic authoring does not necessarily need to literally follow Egri's description of the authoring process and use a simulation-based approach. Any approach can be used as long as it chooses the character actions that those characters would themselves have chosen if they were to autonomous pursue their goals and desires. That is, in the final product that is the story, the audience must be able to perceive each character as being believable. Planning algorithms that use search to generate plans

[4] By extra-diegetic intervention, Egri means the author imposing her will on the actions of the characters. In the case of story generation, the author is a computational process that may employ a drama manager that is distinct from the decision-making processes of any one character that makes decisions about what actions a character performs.

(see Section 3.1.2.1) can be used to computationally formalize the authoring process. The applicability of search-based planning is due to the role that backtracking plays in the authoring process proposed by Egri. The difference between Egri's conceptualization of the process and the way in which a search-based planner functions is that, instead of backtracking to the very beginning of the story every time a dead-end is found, a least-commitment planner backtracks to the last decision point and makes a different choice. The maximum amount of similarity is preserved from one attempt to find a complete story to the next. The concept of premise is also supported by planning. The premise of a story is broken into three parts: initial conditions including character definitions, story outcome, and conflict. The initial conditions and outcome are initialization parameters into the planner. The outcome is the goal of the planner and can be a partial description of the state of the story world after the story is finished or some desired world state that a particular story world character desires to achieve. Conflict is an action and a counteraction which can be expressed as operations in a plan or as goals for the planner to achieve. More detail on how premise is represented in planning is in Section 7.1.1.

3.1.2.1. Planning as a Computational Formulation of the Model of Dramatic Authoring

Planning is used in many story generation systems. For character-centric systems, planning is applied to the story world characters as a method for characters to achieve their individual goals. Since each character is executing a distinct and separately formulated plan, there exists the possibility of conflicts in which a character plan changes the world in ways that cause other characters' actions to fail or consumes a resource that another character's plan requires. Various systems handle conflicts and failures differently, but none predict conflict ahead of time, causing the way the story unfolds to be unpredictable. Only in an author-centric system can planning be considered a computational model of authoring.

An author-centric system that uses planning treats the premise of the story as a problem to be solved. I use the term *story planner* to refer to an author-centric system that models the process of story generation as planning. Examples of story planners are Universe (Lebowtiz, 1984; Lebowitz, 1985) and Tailor (Smith & Witten, 1991). While Mexica (Pérez y Pérez & Sharples, 2001) uses planning as part of its process of generating stories, the generation process is primarily driven by case-based reasoning. Story planners model the process of story generation as problem solving. There is a problem that needs to be solved and it is solved by incrementally changing the story world. Since a story planner is the author of the story and not directly part of the story world, the world can only be changed through the actions of story world characters. Consequently, actions are chosen such that they achieve a particular outcome – a partial description of what the story world should look like at the end of the story. The story goal does not necessarily have to be a world state that one or more story world characters wish jointly or separately to achieve.

There are two reasons why planning makes a good computational model of authoring. First, a plan makes a good model for a fabula. The fabula of a story, as defined by structurist narratology, consists of the literal events – acts and happenings – that happen in the story world between the time the story begins and the time the story ends. The events in the fabula are temporally ordered according to the chronological order that they occur. The fabula can consist of events that are ultimately not told to the audience and in this regard the fabula is relatively complete. A plan is a temporally ordered sequence of operations. An operation is anything that changes the world including character actions (acts) and happenings. The operations in a plan are temporally ordered because in order for a plan to be sound – execution of the plan in the absence of uncertainty guarantees that the goal state will become true – certain causal relationships between operations must hold. That is, for an operation to be executed successfully, certain conditions must be true in the world and these conditions are either true before the plan starts or are caused to be true by previously executed operations. A plan can be partially ordered, meaning that the temporal relationships between certain operations in the plan are unspecified. The definition of a partially ordered plan is as follows.

> **Definition 3.1 (Partially ordered plan).** *A partially ordered plan is a tuple,* $\prec S, O \succ$*, such that* S *is a set of plan steps – operations – and* O *is a set of temporal orderings of the form* $s_1 < s_2$ *where* $s_1, s_2 \in S$.

Partially ordered plans better model fabulas where events can occur simultaneously. The match between plan and fabula is not perfect however. James Allen (1983) enumerates thirteen possible temporal relationships between two events. Plan operators are discrete and are assumed to occur instantly meaning that two operators that are unordered with respect to each other can actually be in any of the thirteen relationships.

The second reason planning makes a good model of story authoring is because the nature of modern planning algorithms, modern partial order planning (POP) algorithms in particular, ensure that stories are coherent. Partial order planning algorithms such as UCPOP (Penberthy & Weld, 1992) explicitly represents the causal relationships between plan steps. A partial order planning algorithm that represents the causal relationships between plan steps is referred to as a partial order, casual link (POCL) planning algorithm.

> **Definition3.2. (POCL Plan).** *A partially ordered, causal link plan is a tuple,* $\prec$S, B, O, L$\succ$, *such that* S *is a set of plan steps,* B *is a set of binding constraints on the parameters of the steps in* S, O *is a set of temporal orderings of the form* $s_1 < s_2$ *where* $s_1, s_2 \in$ S, *and* L *is a set of causal links of the form* $\prec s_1$, p, q, $s_2 \succ$ *where* $s_1, s_2 \in$ S *and* p *is an effect of* s_1 *and* q *is a precondition of* s_2.

POCL ($\prec S, B, O, L\succ, F, \Lambda$)

I. **Termination**. If O or B is inconsistent, fail. Otherwise, if F is empty, return $\prec S, B, O, L\succ$.

II. **Plan Refinement**. Non-deterministically do one of the following.

1. **Goal selection**. Select an open condition flaw $f = \prec s_{need}, p\succ$ from F. Let $F' = F - \{f\}$.
2. **Operator selection**. Let s_{add} be a step that adds an effect e that can be unified with p (to create s_{add}, non-deterministically choose a step s_{old} already in S or instantiate an action schema in Λ). If no such step exists, backtrack. Otherwise, let $S' = S \cup \{s_{add}\}$, $O' = O \cup \{s_{add} < s_{need}\}$, $B' = B \cup$ bindings needed to make s_{add} add e, including the bindings of s_{add} itself, and $L' = L \cup \{\prec s_{add}, e, p, s_{need}\succ\}$. If $s_{add} \neq s_{old}$, add new open condition flaws to F' for every precondition of s_{add}.
3. **Threat resolution**. A step s_{threat} threatens a causal link $\prec s_j, e, p, s_k\succ$ when it occurs between s_j and s_k and it asserts $\neg e$. For every used step s_{threat} that might threaten a causal link $\prec s_j, e, p, s_k\succ \in L$, non-deterministically do one of the following.
   - **Promotion**. If s_k possibly precedes s_{threat}, let $O' = O' \cup \{s_k < s_{threat}\}$.
   - **Demotion**. If s_{threat} possible precedes s_j, let $O' = O' \cup \{s_{threat} < s_j\}$.
   - **Separation**. Let $O' = O' \cup \{s_j < s_{threat}, s_{threat} < s_k\}$ and let $B' = B' \cup$ the set of variable constraints needed to ensure that s_{threat} won't assert $\neg e$.

III. **Recursive invocation**. Call POCL($\prec S', B', O', L'\succ, F', \Lambda$).

Figure 3.2. The POCL algorithm.

A causal link (Penberthy & Weld, 1992) connects two plan steps s_1 and s_2 via condition e, written $s_1 \xrightarrow{e} s_2$, when s_1 establishes the condition e in the story world needed by subsequent action s_2 in order for step s_2 to execute. The POCL algorithm is given in Figure 3.2 (Weld, 1994, fig. 8). The practical implication of the representation of causality is that steps early in the plan cause later steps to be possible. Since a story planner is attempting to solve for a goal – the outcome of the story – the steps that are non-deterministically chosen are those that are necessary and sufficient for the goal to be achieved (Weld, 1994). This implies that story plans are not only coherent, but plot coherent. The main events of the story, as represented by steps in the story plan, are causally necessary for the outcome of the story. Furthermore, there are no plan steps that are not causally related to the outcome of the story. Therefore, all the main events of the story plan have meaning and relevance to the outcome of the story and any story generated by a story planner will be plot coherent.

The fabula of a story consists of all the events that occur in the story world between the time the story begins and the time the story ends. For any reasonably sized story world, the fabula can be potentially quite large. But the entire fabula does not necessarily need to be told to the audience. The sjužet allows for the omission of events in the fabula that are not necessary for the audience to understand the story (Bal, 1997). From the perspective of the audience, the fabula only consists of the events that are told and any additional events that can be inferred from the events that are told. Ryan (1991) argues that every story world is infinitely large and detailed and that it is impossible to represent an entire story world in one's mind. Therefore, a story world is represented not as a complete idea, but as a set of accessibility relationships between a familiar world – the world the audience inhabits – and the story world. The degree to which the story world differs from the world that the audience is familiar with affects how much of the story world the audience must explicitly model and remember. For example, realistic or historical stories are more similar to the world that the audience lives in than science fiction stories. The audience only needs to be told about enough of the fabula to mentally represent the story world through contrast with the real world. Even though the audience need not be told about all the events in a fabula, from the perspective of a storyteller, all events in the fabula can be considered. From the perspective of an author who is creating fiction – creating a fabula – there are infinite possibilities since the story world is assumed to be modeled on a real world.

A plan, as a data structure that contains operations and temporal constraints, is sufficient to capture the complexities of any fabula. A POCL fabula planner, however, only considers operations – events – that change the world in such a way that the outcome of the story is achieved. A POCL planner will not consider the infinity of events that could be happening in the story world while the causally relevant events are taking place. While potentially a limitation, I am interested in generating plot coherent stories and therefore assume that the infinity of causally irrelevant events that could occur in the story world will be eliminated from the sjužet anyway. A simplified fabula is thus considered a favorable quality.

3.1.2.2. Limitations of Planning as a Computational Formalization of the Model of Dramatic Authoring

A story planner is capable of determining whether a set of characters can prove the premise. However, a story planner must have a set of characters to work with that are defined as part of the initial state of the world. To more accurately formalize the model of dramatic authoring, a planning algorithm must also search for a set of characters. Search is a fundamental part of planning and the search for a set of characters that can prove the thesis can be integrated into the story planning algorithm by treating story planning as an open world planning problem. Open world planning is planning under uncertainty. In the case of a story planner, what are unknown are descriptions of the characters in the story world. More detail about how open world planning is used to integrate the search for a set of characters with the search for a story that demonstrates the premise is in Section 3.5.

One thing that is not native to planning and therefore not native to story planners is the notion of character believability. A story planner produces a story plan in which a predefined outcome in the story world is true. The story planner uses characters to effect changes in the story to achieve that outcome and does not necessarily consider whether it makes sense for a character to perform any particular behavior from the perspective of the audience. Thus a story planner is driven by formal cause. A planning algorithm used by a story planner must take into consideration character believability or the story planner will fail to computationally formalize the model of dramatic authoring because story world characters must appear to the audience to react to the environment they are situated in. The way in which character believability is modeled in a story planner is described in Section 3.4.

3.2. A Planning Approach to Story Generation

Given a model of dramatic authoring, the basic problem I address is the design of a story generation system that implements the model of dramatic authoring in such a way that the stories generated have both strong plot coherence and strong character believability. That is, given a premise, the story generation system must "prove the premise" while also considering plot coherence and character believability. The model of dramatic authoring defines this process as a search for a set of characters that can act believably in the story world and achieve the premise of the story.

My approach is an author-centric approach, using planning to search simultaneously for (a) a sequence of character actions that forms a coherent plot and (b) a set of characters that can be part of the story. The Fabulist system implements a planning algorithm that has been modified to be able to reason about possible sets of character, plot coherence, and character believability. The story planning algorithm used in Fabulist implements a particular model of plot coherence. As discussed previously, story planners naturally generate story plans that are plot coherent. The exact nature of plot coherence in stories generated by a story planner is described in Section 3.2.1. Character believability, on the other hand, is a concept that is not explicitly addressed by planning algorithms. The planning algorithm must be modified to reason about character intentions based on a formal model of character believability. The formal model of character believability used by the story planning algorithm in Fabulist is described in Section 3.2.2. Finally, the story planning algorithm used by Fabulist must be able to search for a set of characters that can prove the premise while simultaneously searching for a story that solves for the premise. Section 3.5 discusses how a story planner can simultaneously search for character and plot using open world planning.

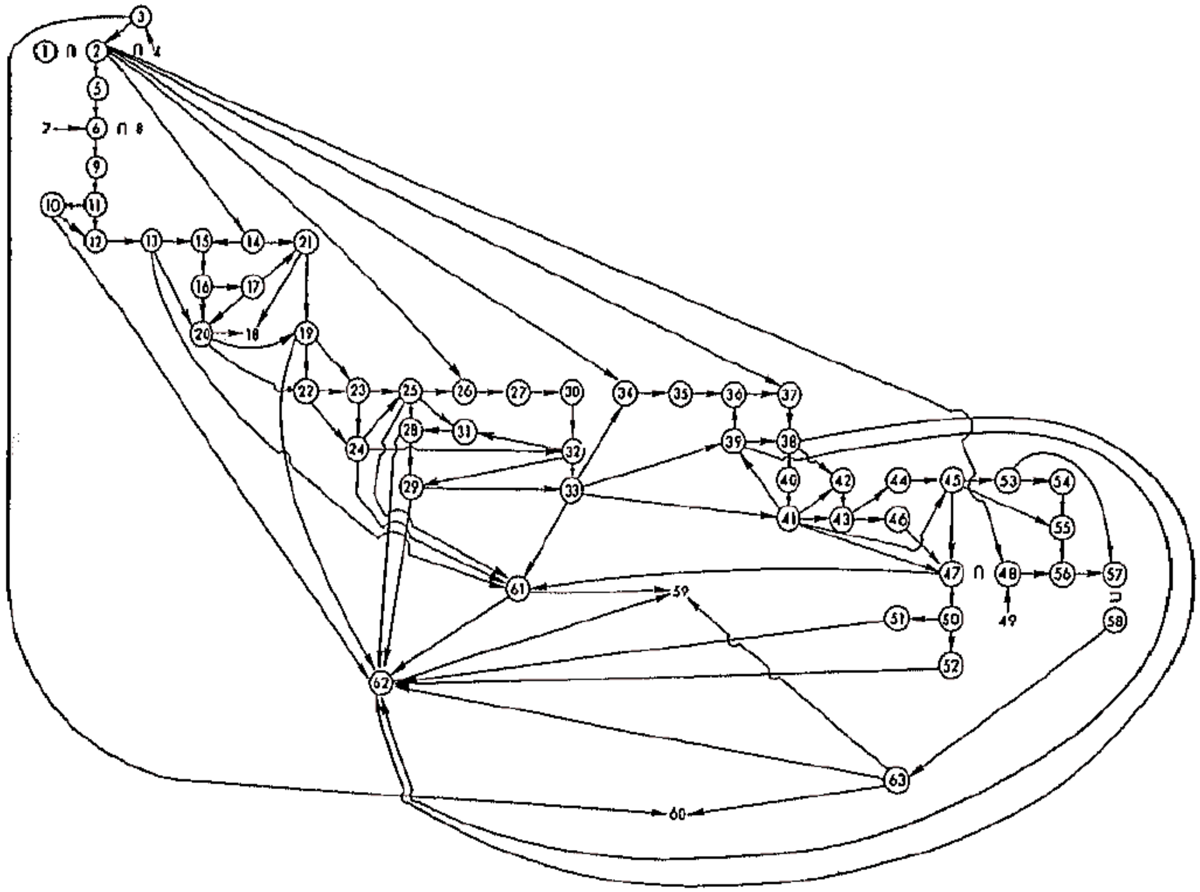

Figure 3.3. A causal network representation for a story.

3.2.1. Model of Plot Coherence

My definition of plot coherence states that a story is plot coherent if the main events in the story appear to have meaning and relevance to the outcome of the story. The definition is somewhat vague in that it does not describe exactly what it means for an event to have meaning and relevance to the outcome of the story. A story planner that formalizes the model of dramatic authoring requires a more precise, computational model of plot coherence. One way an event can be meaningful and relevant to the outcome of the story is if that event is part of a causal chain of events that results in the outcome of the story (Trabasso & Sperry, 1985). A causal relationship exists between two events in a story if one event is necessary in order for another, later event to occur. Necessity is tested by the counterfactual argument of the form: *If not A then not B*. That is, an event A is said to be necessary for event B to occur if it is the case that if A had not occurred, then B could not have occurred (Trabasso & Sperry, 1985). When the causal relationships between all events in a story are captured, a causal network is established. Figure 3.3 (Trabasso & Sperry, 1985, fig. 1) shows a causal network for a story. The numbers correspond to events in the story and the directional links represent causal relationships between events. Circled numbers indicate events that are part of a causal chain that terminates with the outcome of the story. A causal chain is a path from one event to another event or to the outcome of the story in the causal network for a story. If every main events of a story is part of some causal chain that terminates in the outcome of the story, then the story is plot coherent.

Modern least-commitment planners such as UCPOP (Penberthy & Weld, 1992) use a theory of causation to construct sound plans. A sound plan is a plan that is guaranteed to execute without failure in a world in which changes in the world only occur when plan steps are executed. Steps are only included in the plan if they causally link to the goal or to another step that is already in the plan. That is, there is a path from every plan step to the goal of the plan through the directed, acyclical graph (DAG) of plan steps and causal links. Representing the events in a story as plan steps and the outcome of the story as the goal of the plan implies that every event is part of some causal chain terminating in the outcome.

The way in which plan steps are causal linked in a story plan is a literal implementation of the computational model of plot coherence; all story plans are plot coherent. However, using causal linkages to model plot coherence is overly constraining because *all* events in the story have meaning and relevance to the outcome of the story. The definition of plot coherence requires that only the *main* events of a story have meaning and relevance to the outcome of the story. The definition of plot coherence indicates a certain degree of "wiggle room" for the author to include insignificant events (events that are not considered part of the plot) that do not have meaning and relevance to the outcome of the story. With a story planner using the causal link model of plot coherence, all story plans have a restricted form of plot coherence I call story coherence.

> **Definition 3.3 (Story coherence).** *A story is story coherent if all events in the story appear to have meaning and relevance to the outcome of the story.*

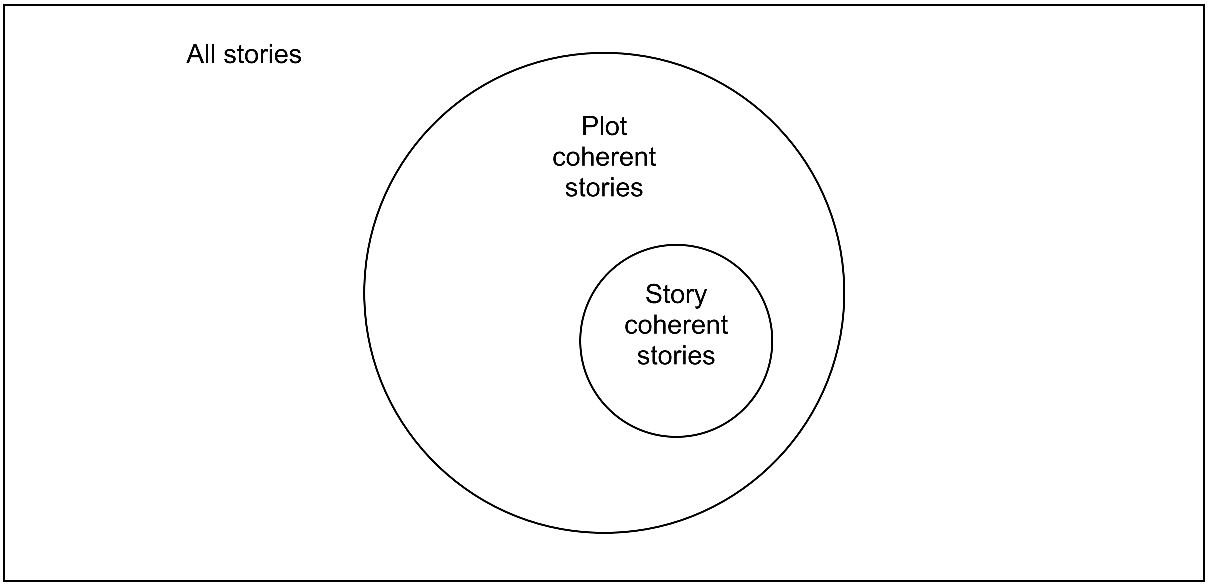

Figure 3.4. The relationship between plot coherent stories and story coherent stories.

Story coherence does not allow for any events in a story that do not have meaning and relevance to the outcome of the story. It is never the case that an event is not part of some causal chain that terminates in the outcome of the story. For example, the story represented in Figure 3.3 is not story coherent, because there are events (e.g. 18, 49, 59, and 60) that are not part of any causal chain that terminates with the outcome of the story.

Story coherent stories are also plot coherent; the only distinction is whether or not "wiggle room" is tolerated. Of all possible stories that can be told, a subset of the stories is plot coherent. Of all possible plot coherent stories, a subset of the stories is story coherent, as shown in Figure 3.4. Plot coherence is a constraint on the form that stories can take that is assumed to be favorable to the audience. Story coherence further constrains the form that stories can take. However, there is no indication that story coherent stories are more favorable than plot coherent stories. The causal link model of plot coherence limits the stories that a story planner can generate to those that are story coherent. There are some stories – those that are plot coherent but not story coherent – that a story planner cannot generate. The nature of the limitation guarantees that all stories generated by a story planner are plot coherent. The limitation is acceptable because plot coherence can be guaranteed without making any modifications to the planning algorithm that increase the complexity of the planning algorithm or interfere with the completeness of the algorithm or the soundness of the solution plans; causal relationships are established between plan steps as a side effect of the planning algorithm itself.

3.2.2. Model of Character Believability

A story has character believability if the actions of the characters in the story appear to be motivated by the characters' internal traits and desires. Like plot coherence, the definition of character believability is somewhat vague in that it does not describe exactly what it means for a character action to be believable. Unlike plot coherence, however, there is no structure or process native to least-commitment planning that can be used to computationally model character believability or even a limited form of character believability. Character believability refers to the numerous elements that allow a character to achieve the "illusion of life" (Bates, 1992). To be of practical use to a story planner, a more precise definition of character believability is required: one that can be computationally modeled. Therefore, I define a believable character as one whose actions are consistent and intentional, as shown in Figure 3.5. Character consistency means that the actions that a character performs are consistent relative to some model of the traits that the character possesses. That is, given a description of the character and its traits, the actions the character performs are a direct and observable manifestation of those traits. A consistent character will not perform any action that leads the audience to believe that the character has an internal trait that is contrary to the actual traits the character is defined to have. Character intentionality means that the actions that a character performs are motivated by a desire to achieve some goal. That is, character

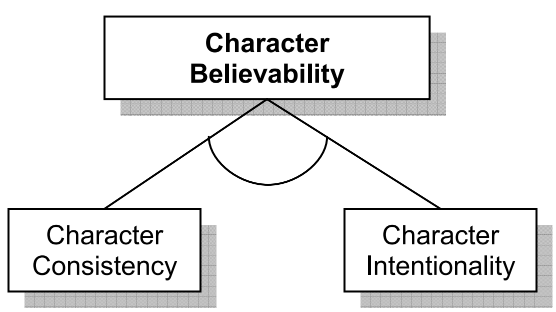

Figure 3.5. Character believability broken down into its constituent components.

actions are not performed without reason. Characters appear to have intentions to achieve goals and the actions that the characters perform in the story world appear to be driven by those intentions.

There is a third component of character believability: appearance. Appearance has to do with the physical description of a character and the way in which the character moves. The appearance of a believable character is handled during execution of a story, where execution means rendering into written or spoken form or rendered into a cinematic performance by animated agents in a virtual world[5]. Execution of believable characters is addressed by (Blumberg & Galyean, 1995; Maes et al., 1995; Perlin & Goldberg, 1996; Loyall, 1997; Hayes-Roth, van Gent, & Huber, 1997; Lester et al., 1999) and will not be considered further here.

Character consistency and character intentionality are concepts that can both be readily applied to planning. Consistency implies some sort of character trait model and previous systems such as Universe (Lebowitz, 1984) have adapted planning to utilize character trait models. Chapter 5 discusses how the least-commitment planning algorithm can be adapted to account for a trait model of personality. Intentionality implies purposeful action directed towards the achievement of goals; planners specialize in finding sequences of actions that achieve goals. The author-centric approach uses planning to solve for goals that represent an outcome or premise that the author wishes a story to have. However, the author's goals are meaningless if the story world characters do not achieve those goals in believable way. One technique would be to use heuristic evaluation functions to prune portions of the search space in which characters do not appear to act believably according to some metric. Such an approach will increase the likelihood that a story plan has character believability, but it will not guarantee that the story planner will always be able to find a story plan that is not rejected by such a heuristic. Instead, a story planner must represent the characters' goals in addition to the author's goals and solve for both so that characters are believable and the outcome or premise of the story is achieved. Chapter 4 discusses how the least-commitment planning

[5] Character-centric systems often do not distinguish between generation and execution since story emerges from the real-time performances of the characters.

algorithm can be adapted to account for both the author's goals and the characters' goals in order to ensure character believability.

3.2.3. Open World Planning for Story Generation

The search through the space of sets of characters is already partially handled by planning algorithms. When a character action is chosen for the story plan, the character that performs that action is, by definition, part of the story. If the story does not prove the premise because of a choice about which character performs a certain action, the planner backtracks to the relevant decision point and chooses an alternative character. However, the model of dramatic authoring suggests that characters should be crafted based on what needs to be accomplished in the story. If the crafted characters do not cause the premise to be proved, then new characters must be crafted. Search through the space of sets of characters implies that if a story plan does not prove a premise, then the planner backtracks and chooses a new formulation of that character.

In order to avoid the inefficiencies of backtracking to the first decision point every time, the planner can take a least-commitment approach to character. In a least-commitment approach to character, each character begins without any formal specification of character traits. As characters express certain traits through the character actions they are assigned to perform, the planner commits to defining that character with that specific trait or set of traits. If the commitment creates an inconsistency, such as expressing contradictory traits in a character, the planner backtracks to an earlier decision point and either chooses a different character to perform the action, a different action for the character to perform, or a different definition of the character in question. The reason such an approach works is because character is a perceived attribute. The audience is not aware of any aspect of a character until it is expressed in an observable manner. Therefore, the planner does not need to commit to any one aspect of any character until the character needs to express it for the purpose of finding a story that proves the premise. If the planner makes the wrong commitment, then the planner can backtrack and make a different choice later on. Searching the space of possible sets of characters while concurrently searching for a sequence of character actions that makes a story is similar to the notion of open-world planning. In open-world planning, the initial conditions of the world are not known with any certainty and the planner must discover the true conditions of the world in order to complete the plan. The complete definition of each character is not known in the initial state of the story world. During the process of plan construction, the definitions of the characters become known as the planner works to find a fabula plan. Chapter 6 discusses how open world planning can be used to search for a set of characters while concurrently searching for a sequence of character actions that proves the premise.

3.3. Summary

This chapter discusses the relationship between search-based planning technologies and narrative. I present a model of dramatic authoring that describes the way in which an author can go about ensuring that the authoring process delivers a product – a story – that has meaning. I present two computational formalizations of the model of dramatic authoring. The first is a character-centric approach and the second is an author-centric approach that uses search-based planning. Similarities between the model of dramatic authoring and the POCL planning algorithm suggest that planning is a good candidate for a computational model of dramatic authoring.

A Planning approach to story generation requires a new type of planning algorithm that implements computational models of plot coherence and character believability. A limited form of plot coherence, called story coherence, is a side-effect of the POCL planning algorithm. A story coherent narrative is one in which all events (not just the main events) have meaning and relevance to the outcome of the story. Since a planner only inserts actions into a plan that are necessary and sufficient for the goal to be achieved, all actions in the story are part of some causal chain that terminates in the outcome of the story. Character believability is more problematic for planners. I describe character believability as being made up of two attributes: consistency and intentionality. That is, for a story world character to behave believably, it must act in a way that is internally consistent to some definition of personality and act as if it were committed to achieving goals.

Finally, for a planning algorithm to be considered a computational model of dramatic authoring, it must be able to search not only for a sequence of events that achieves the story goal, but also to search through the space of possible characters. This implies that the planner should take a least-commitment approach to determining what the traits of the characters are while determining what the characters should do. Not only does this alleviate the need for the human author to fully specify the characters up front, but also allows the planner to simultaneously determine plot and character in a form of open-world planning.

Chapter 4

Intent-driven planning

The definition of character believability in this work is constrained to focus on consistency and intentionality of character behavior in the story world. Intentionality refers to the way in which characters should be observed by the audience to have goals and act to achieve those goals. It is not sufficient, however, for a character to act intentionally if the audience is not capable of inferring that character's intentions from the circumstances that surround the character in the story world. Characters should be committed to their goals in a rational fashion and also be observed to react to the world around them and to opportunistically form these goals as the story unfolds. The audience of a story is not a collection of passive observers. Instead, the audience actively performs mental problem-solving activities to predict what characters will do and how the story will evolve (Gerrig, 1993). One way to generate a story that facilitates the audience's problem-solving activities is to simulate the intention recognition process of a hypothetical audience in order to inform the plan search process so that solution plans demonstrate character intentionality in a way that is understandable to the audience.

In this chapter, I present a novel planning algorithm, the intent-driven partial order causal link (IPOCL) planner, that generates fabula plans in which characters act intentionally and in which that intentionality is observable. Instead of relying on heuristics to focus plan search on portions of the plan space in which characters appear to act intentionally, I argue that the conventional partial order planners are unable to generate observable intentionality. I present an expanded representation of a plan that enables a fabula planner to access a potentially larger search space in which all characters appear to act intentionally.

4.1. Planning for Rational Agents

The issue of intentional agent behavior is of concern to researchers working towards building agents that are capable of practical and rational action. One relevant approach to rational agents is an Belief/Desire/Intention (BDI) architecture such as (Bratman, 1987; Bratman, Israel, & Pollack, 1988). The BDI architecture directly represents the relationship between an agent's beliefs, desires, and intentions. The BDI architecture is paired with underlying formalisms that specify how beliefs and desires are transformed into intentions to act. Beliefs are statements about the world in which the agent is situated that the agent believes to be true. Desires are partial descriptions of future world states that the agent would like to see true. Intentions are desires that the agent commits to. Unlike desires, which do not need to be consistent with each other, the set of desires that are transformed into intentions must be compatible with each other. Intentions are structured into plans which the agent intends to execute in order to bring about the intended world state. Central to the idea of BDI agents is the observation that a rational agent is committed to its intentions and its plan to achieve its intentions (Bratman, 1987; Bratman, Israel, & Pollack, 1988; Cohen & Levesque, 1990).

An agent built on a BDI architecture reasons until it determines the next action that it should take in the environment to move closer to achieve its intended world state. In most circumstances, it is not practical or realistic for an agent to perform unbounded reasoning. Agent reasoning should be resource-bounded for an agent to be practical (Bratman, Israel, & Pollack, 1988). A commonly bounded resource is time, meaning that an agent only has a finite amount of time to make any determination. The BDI architecture of (Bratman, Israel, & Pollack, 1988) is bounded from taking indefinite resources by assuming that an agent has a library of recipes – partially constructed plans – that are believed to achieve certain world states. One recipe is selected to be instantiated as the executable plan for achieving the agent's intention. Agent operation is cyclical. Beliefs and desires are transformed into intentions that are structured as plans. The plan is executed, although observation is interleaved with execution so that the agent is able to opportunistically adjust its beliefs and desires. The cyclical nature of operation ensures that an agent can opportunistically respond to unplanned changes to the environment.

BDI architectures strike a practical balance between reactive and deliberative planning (Bratman, Israel, & Pollack, 1988) which is necessary for a rational agent to operate effectively in a dynamic and unpredictable environment. The environment may be unpredictable due to the presence of other agents that are acting deliberately to achieve their own individual intentions, in which case, the environment can change drastically between the time an agent plans and the time an agent acts. Norling and Sonenberg (2004) argue that the BDI architecture can be adapted to modeling believable characters. Fabula planning, however, does not require reactive agents since the fabula planner has complete knowledge of the story world. Furthermore, no agent can act to change the world without that action being explicitly included as part of the fabula plan. What the BDI architecture facilitates is intentional behavior; BDI agents can be seen to commit to intentions based on their beliefs about the world around them and can be seen to act deliberately to achieve those intentions. For a fabula plan to successfully capture the believability of story world characters, those characters must be seen to act intentionally. While BDI agents or the formalisms behind the BDI architecture do not come into play in story *planning*, the fabula plans that are generated must show characters acting as if they had beliefs and desires and were capable of forming and deliberately pursuing intentions.

4.2. Mismatch between Conventional Planning and Fabula Planning

Planners are means-ends tools for problem-solving. In the case of fabula planning, the problem that is being solved is the generation of a story as an ordered (partially or otherwise) sequence of character behaviors. Planning is conventionally used as part of a larger process, such as the BDI architecture (Bratman, 1987), wherein a goal is selected and committed to and a plan for achieving that goal is created or retrieved from a library. The conventional use of planning makes the following assumptions.

- The plan is created by or for a single agent (Bratman, Israel, & Pollack, 1988), or for a collection of cooperating agents (Grosz & Sidner, 1990).
- The goal state of the planning problem describes properties of the world that are intended by all agents that are to execute the plan.

Because the planner assumes that the goal state is an intended world state, the intentions of the agent are encoded implicitly as part of the planning problem. The SharedPlans (Grosz & Sidner, 1990) formalism addresses the situation where more than one agent collaborates to construct a joint plan for achieving some goal. SharedPlans addresses the cases where all agents intend that a joint goal is achieved or where one agent has a goal and "contracts out" part of the task to another agent by communicating its intentions (Grosz & Kraus, 1996). SharedPlans handles the coordination of many individual plans into a single joint plan by defining how agent intentions to perform actions, and agent intentions that goals and sub-goals be achieved, constrain the behaviors of the individual agents. SharedPlans, however, does not handle the situation where agents may not be cooperating and thus have distinct and different goals.

The success of a story is partially reliant on the believability of the story world characters. Specifically, story world characters must be seen to act intentionally towards individual and distinct goals. The use of planning for fabula generation makes the following assumptions.

- The plan is created for multiple agents that are not necessarily cooperating.
- The goal state of the planning problem describes properties of the world that are not necessarily intended by any of the agents that are to execute the plan.

Story world characters must be able to have individual goals that are possibly distinct from the goal state of the planning problem. However, the goal state cannot be a conjunction of the individual goals of the story world characters either because this would imply cooperation. Cooperation in turn requires that the characters have consistent goals, meaning one character's goal cannot negate another character's goal. The assumption of cooperation can be overcome if the planner distinguishes the goal state of the planning problem –

partially describing the state of the story world when the fabula plan has completed execution – and the individual goals of story world characters.

> **Definition 4.1 (conventional planning problem).** *A conventional planning problem is a tuple,* $\prec$I, G, A, Λ$\succ$, *where* I *is a set of atomic ground sentences that completely describe the state of the world before plan execution (time* t=*0),* G *is a set of atomic ground sentences that partially describe the state of the world after execution is complete (time* t=*∞),* A *is a set of agents that are cooperating to achieve* G*,* Λ *is a set of action schemata that describe the actions that agents in* A *can perform in the world, and* $G = \bigcup_{a \in A} G_{t=0}(a)$ *where* $G_{t=0}(a)$ *is the set of goals committed to by agent* a ∈ A *before the beginning of plan execution.*

The purpose of the planner is to construct a plan that makes *G* true in the world, given *I*. In effect, the assumptions of conventional planning limit the space of plans that the planner can search to the set of plans in which the goal state is committed to by all agents in *A* because no agent can not intend *G*. After all, under the assumption of cooperation, G is the conjunction of all of the agents' goals. The assumptions of fabula planning are a generalization of the assumptions of conventional planning in that they relax some of the constraints placed on the planning problem.

> **Definition 4.2 (fabula planning problem).** *A fabula planning problem is a tuple,* $\prec$I, G, A, Λ$\succ$, *where* I *is a set of atomic ground sentences that completely describe the state of the world before plan execution (time* t=*0),* G *is a set of atomic ground sentences that partially describe the state of the world after execution is complete (time* t=*∞),* A *is a set of agents,* Λ *is a set of action schema that describe the actions that agents in* A *can perform in the world.*

The goal state, *G*, is referred to as the *outcome* of the fabula plan. The fabula plan represents the story about how the initial world state is transformed by the story world characters into the world state in which the outcome holds. There are two subtle distinctions between the fabula planning problem and the conventional planning problem. First, the goal state can be distinct from the individual goals of all the agents in the world. That is, it is not the case in the fabula planning problem that $G = \bigcup_{a \in A} G_{t=0}(a)$, meaning that the goal of the planning problem is not necessarily a conjunction of the goals the agents have in the initial state of the world. Furthermore, it is possible that $G \neq \bigcup_{a \in A} G_{0 \le t < \infty}(a)$ where $G_{0 \le t < \infty}(a)$ is the set of goals committed to by an agent $a \in A$ over the time span of the story. That is, even if there are agents that have not committed to achieve any goal at $t = 0$, the outcome of the story is not necessarily the same as the set of goals the agents have committed to by the time the story ends ($t = \infty$). The practical consequence of the definition of the fabula planning problem is that none of the sentences that describe the story outcome need to be intended by any of the story world characters at any time.

The second distinction between the fabula planning problem definition and the conventional planning problem definition is that the fabula planning problem does not necessarily encode any information about the individual goals of the agents. In conventional planning, the individual goals of the agents are encoded as inputs into the conventional planner because $G = \bigcup_{a \in A} G_{t=0}(a)$, implying that an agent's intentions are known before the plan begins. In fabula planning, the equality does not hold. The implication of the definition of fabula planning is not that the agents have no intentions. Rather, the intentions are not specified as inputs into the planner. A fabula planner that considers character intentionality in addition to author intentionality should search the space of all plans in which the agents have individual goals that are possibly distinct from the goal state of the planning problem *and* that the individual goals of the agents are not necessarily formed before the initial state of the plan. Thus for any agent $a \in A$, $G_{0 \le t < \infty}(a)$ is the set of goals that agent a has committed to during the (inclusive) interval between the initial world state, *I*, and the goal state, *G*. Agent intentions can be encoded as part of the initial state of the planning problem as special atomic ground sentences of the form, `(intends a ga)`, signifying that agent *a* intends that g_a become true. There is no guarantee, however, that the solution fabula plan that is found by the planner will entail agent *a* acting on the intention. Since the fabula planning problem definition is a generalization of the conventional planning problem definition, the space of plans searched by a fabula planner is not smaller than the space of plans searched by a conventional planner.

The fabula planner described in the remainder of this chapter searches the space of plans in which individual agent goals are distinct from the goal of the planning problem and in which agents are not cooperating. Agents can either be given intentions as part of the specification of the initial world state or develop them during the course of the plan. The IPOCL planning algorithm accomplishes this by expanding the representation of the plan structure to generate information about the intentions of the individual agents. Algorithmically, IPOCL simultaneously searches the space of plans and the space of agent intentions. Agent intentions are ensured to be plausible through the invocation of an intention recognition process that simulates the problem-solving capabilities of a hypothetical, active audience.

4.3. Intention Recognition in Fabula Planning

Plan recognition is the process of inferring an agent's plan given a set of observed agent behaviors. Intention recognition is related to plan recognition except that the purpose is to infer the agent's intention or goal. The basic inference process starts with a set of goals that an agent might be expected to pursue in the domain and one or more observed actions performed by that agent. The task is to determine how those observed actions contribute to those goals. Systems are traditionally provided with a set of actions that the agent might execute in the domain and a library of recipes that encode how an agent might go about performing these actions (Carberry, 2001).

Gerrig (Gerrig, 1993) demonstrates that a story's audience is not passive. Specifically, Gerrig claims that the audience actively performs problem-solving in order to predict the outcome of the story and the fate of story world characters. This problem-solving involves interpreting character actions, inferring future events and the probability of favorable outcomes (e.g. the low probability of a favorable outcome invariably leads to feelings of suspense in the audience (Gerrig and Bernardo, 1994)). Since the audience is not aware of the fabula beyond the point observed during the telling of the fabula, the audience is only

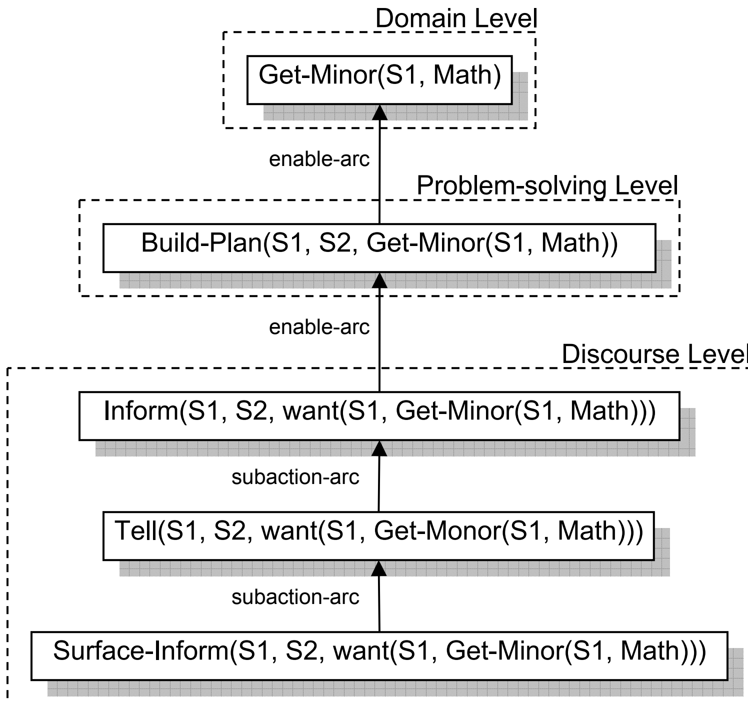

Figure 4.1. Example of a tripartite model for dialogue.

able to draw upon the visible actions that story world characters perform as well as happenings in the story world as a source for their problem-solving. In short, the audience actively applies plan and intention recognition towards the story world characters. It makes sense, therefore, for a fabula generator to include structures that can be used by the audience to make inferences about the story. By doing this a fabula generator will be able to generate stories that support the active problem-solving that will be performed by the audience. Specifically, the story will support the inferences that the audience will make about the intentions of the story world characters.

Lambert and Carberry (1991) present a tripartite model of dialogue that is used for intention recognition. The model segregates a dialogue plan into three levels: discourse, problem-solving, and domain. Lambert and Carberry's model is designed to handle intention recognition of a single agent (a reasonable assumption for a discourse plan), and as such has a single goal that is intended by the single discourse-generating agent. The tripartite model distinguishes between domain, problem-solving, and communicative actions in a way that preserves the relationships among them. The discourse actions are actions taken to achieve communicative goals that the agent might have. From the discourse actions, one can infer that the agent has one or more problem solving actions which, when executed, caused the agent to form discourse goals and plan discourse actions. Problem solving actions exist on a higher, problem-solving level. From the problem solving actions, one can infer that the agent was performing problem-solving in order to carry out domain actions that achieve a domain goal – a description of an intended world state. Domain actions exist on the highest level: the domain level. An example of a tripartite model that demonstrates the relationship between the level hierarchy is shown in Figure 4.1 (Lambert & Carberry, 1991, fig. 2).

One can imagine there being a fourth, implied level above the domain level, which one might call the *intention level*. The intention level contains a set of intentions that the agent is committed to. In the domain of discourse processing, the intention level might only contain a single intention that is directly related to the goal in the domain level. As a discourse model, Lambert and Carberry's tripartite model is not appropriate for fabula planning because it only addresses a single agent with a single, inferred (because the model does not include an intention level) intention. In a fabula plan involving many characters, each character can have many intentions that cannot be represented by the physical and mental actions in the story world domain.

IPOCL utilizes two representational levels: the domain level and the intention level. The domain level contains physical and mental character actions of which all causal and temporal relationships are represented. The intention level captures the commitments of each story world character to achieve internal character goals. The relationship between the domain level and the intention level captures the relationship between the actions that characters perform in the story world to the characters' internal goals. In the domain level, a causal link (Penberthy & Weld, 1992) connects two plan steps s_1 and s_2 via condition e, written $s_1 \xrightarrow{e} s_2$, when s_1 establishes the condition e in the story world needed by subsequent action s_2 in order for step s_2 to execute. The fabula plan consists of sequences of character actions that are *intentional*. Intentional character actions are actions in the domain level that are related to structures that record that character's intentions in the intention level. All character actions must be intentional and each character has its own intentions. Each sequence of intentional character actions is referred to as an interval of intentionality.

Definition 4.3 (Interval of intentionality). *An interval of intentionality is a tuple,* $\prec$S, a, g_a, $s_f\succ$*, such that* S *is a set of plan steps in a plan* P = $\prec S_P$, B, O, L$\succ$, a *is a character agent such that all steps in* S *are performed by character* a, g_a *is an internal character goal held by* a*, and* $s_f \in$ S *– referred to as the final step of the interval – has* g_a *for one of its effects and all other steps in* S *temporally precede* s_f *in the step ordering* O *of plan* P.

The interval of intentionality is the set of actions that character *a* performs to achieve the internal character goal, g_a. An interval of intentionality roughly equates to the notion of a *Full Individual Plan* (FIP) in the SharedPlans formulation (Grosz & Sidner, 1990). A full individual plan is a portion of the larger *Full Shared Plan* (FSP) that a single agent is responsible for executing. The distinction between a fabula plan and an FSP is that the full fabula plan is not made up of many individual FIPs generated by collaborating planning agents. Instead, a fabula plan is constructed as a whole and the individual character actions that make up the whole plan are annotated as to what intention they might be used to achieve.

An interval of intentionality can contain more than one step with g_a as an effect. This is necessary in the case where another action undoes g_a in the world and the condition must be reestablished. By definition, internal character goals partially describe a world state that the character commits to achieving. Commitments persist through time and a character will remain committed to the goal even though the desired world state is undone (Bratman, 1987). IPOCL does not explicitly represent the release of a commitment except to say that the interval of intentionality is bounded and character actions that occur prior to or after the character's interval of intentionality can conflict with the character's internal goal since, by definition, that character is not committed to the goal at those points.

Structurally, within the fabula plan, an interval of intentionality is represented as part of a *frame of commitment*, which is a data structure recording the commitment that a story world character has to achieve some internal character goal.

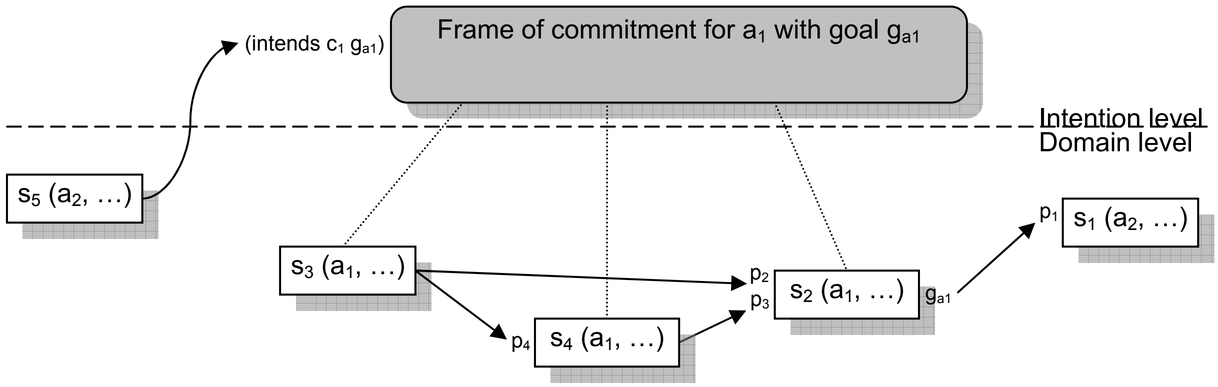

Figure 4.2. An IPOCL plan with a single frame of commitment and motivating step.

> **Definition 4.4 (Frame of commitment).** *A frame of commitment is a tuple,* $\prec$a, g_a, I$\succ$, *where* a *is a character agent,* g_a, *is the internal character goal that* a *is committed to, and* I *is an interval of intentionality which shares the same character and internal character goal with the frame of commitment.*

The interval of intentionality, as a component of a frame of commitment, describes the interval of the fabula plan during which character *a* is committed to g_a. A frame of commitment resides in the intention level of the fabula plan. The purpose of the frame of commitment is to record a character's internal character goal, g_a. However, from the perspective of the audience, it is not enough to declare a character as having a goal; in order to make inferences about character intentions and plans, the audience must observe the characters forming and committing to goals (Gerrig, 1993). Therefore, each frame of commitment is associated with a condition, e_g, of the form `(intends a ga)`, which indicates that for a character to commit to an internal character goal, *c* must intend to bring about that world state. The condition, e_g, is established in the world by some plan step that has e_g as an effect. That is to say, something in the world *causes* character *a* to commit to g_a. The plan step that causes e_g and consequently causes the frame of commitment is referred to as the *motivating step* for the frame of commitment. The motivating step necessarily precedes all plan steps in the frame of commitment's interval of intentionality. See Figure 4.2 for a representation of an IPOCL plan with a single frame of commitment and a motivating step for that frame.

The purpose of the extensions IPOCL makes to the conventional POCL planning algorithm is to ensure that all story world characters that participate in a fabula plan appear to act believably with respect to intentionality. That is, story world characters act when they are committed to some internal character goal. To satisfy this requirement, all character actions in an IPOCL plan must be intentional in the final solution plan. Actions do not need to be part of an interval of intentionality in intermediate plan nodes.

> **Definition 4.5 (Intentionality).** *An action in plan P is intentional if it belongs to some interval of intentionality that is part of a frame of commitment in P. The intended purpose of this action is as part of a causal chain that terminates in an action – a final step – that achieves the internal character goal of that frame of commitment (As a short-hand, I refer to an action that is part of an interval of intentionality as being part of the interval's frame of commitment, since a frame of commitment has exactly one interval of intentionality).*

Actions that are not intentional, according to Definition 4.5, do not belong to any interval of intentionality. Such unintentional actions are referred to as *orphans*. In order for an IPOCL plan to be considered complete, all actions (except for a special class of actions called happenings – see Section 4.4.1.1) must be part of at least one frame of commitment. A character action can belong to more than one interval of intentionality as described in Section 4.4.3.1. An intentional character action can be part of a deliberative sequence of action or the consequence of a spontaneous decision. A deliberative agent has an interval of intentionality that encapsulates a sequence of character actions that eventually lead to a character goal being achieved. A wanton agent reactively forms goals and acts immediately, as if acting on a whim; an interval of intentionality for a wanton agent may be very short, possibly even encapsulating a single action. For both deliberative behavior and wanton behavior, a motivating step has the effect of causing the agent to form an intention and to act to achieve that intention. Both deliberative behavior and wanton behavior can be differentiated from random behavior where the agent performs actions that are not motivated by any intention. Such random behaviors would be orphans in the solution story plan.

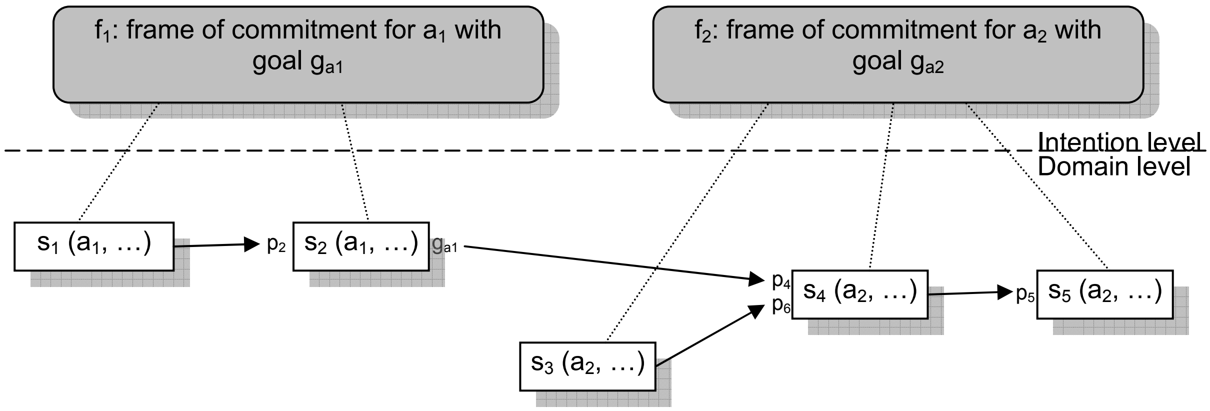

Figure 4.3. An IPOCL plan with two frames of commitment where one frame is in service of another.

Many times there are interactions between the intentions of a single character or between different characters. If the effect g_c of the final step s_f of a frame of commitment f_1 causally establishes a precondition of another step s_2 in another frame of commitment f_2, then f_1 is *in service of* s_2. Consequently, f_1 is also in service of f_2. Figure 4.3 shows a fabula plan with two frames of commitment where one frame is in service of another frame.

IPOCL expands the typical POCL plan representation to include frames of commitment. This is necessary so that a plan can have character goals that are possibly distinct from the goal of the planning problem. Since character goals can be committed to at any time during the fabula plan, internal character goals are not encoded in the planning problem definition. Instead they are encoded in the plan itself as frames of commitment. The definition of an IPOCL plan is as follows.

> **Definition 4.6 (IPOCL plan).** *An IPOCL plan is a tuple,* $\prec$S, B, O, L, C$\succ$, *where* S *is a set of plan steps,* B *is a set of binding constraints on the free variables in the steps in* S, O *is the set of ordering constraints on the steps in* S, L *is a set of causal links between steps in* S, *and* C *is a set of frames of commitment.*

The sets, *S*, *B*, *O*, and *L* are defined in the standard way (Penberthy & Weld, 1992). The frames of commitments in *C* are defined in Definition 4. The definition of plan completeness is given as follows.

Definition 4.7 (IPOCL Plan completeness). *An IPOCL plan is complete if and only if (1) all preconditions of all plan steps are established, (2) all casual threats are resolved, and (3) all plan steps are intentional.*

Conditions 1 and 2 together make up the conventional definition of plan completeness, which can be termed *causally complete*. A fabula plan in IPOCL can be causally complete without being fully complete under Definition 4.7. When a plan is causally complete but not fully complete, then the plan contains orphans. If there are no ways to correct for the orphans, IPOCL backtracks to find another possible complete solution plan.

4.4. Integrating Intention Recognition into Least-Commitment Planning

Frames of commitment are products of the simulation of a hypothetical audience's process of intention recognition applied to story world characters. The simulated intention recognition process is integrated into the IPOCL algorithm such that IPOCL simultaneously searches the space of partially ordered plans in which character agents have goals distinct from the story outcome and the space of character agent intentions. Leaf nodes of the search space are either complete plans (under Definition 4.7) or plans with inconsistencies. Internal nodes are incomplete plans in that they have one or more flaws. A flaw is a decision-point and the children of any internal node are distinct refinements of the parent plan, repairing a single flaw. Partial-order planners represent flaws explicitly as annotations of the plan. For example, an open condition (Penberthy & Weld, 1992) is a plan flaw in which a precondition of a plan step is not causally established by a preceding step or the initial world state. New plan steps are instantiated in a backward-chaining fashion with the sole purpose of establishing open conditions. However, for character actions to appear intentional, every character action in the story plan must be part of the interval of some frame of commitment. Thus, when a plan step is newly instantiated, it must be declared part of the interval of an existing frame of commitment or a new frame of commitment must be created that describes a possible intention that the character has for performing that action. Either way, every plan step in a complete plan is linked to a frame of commitment and is thus declared intentional.

In addition to open condition flaws, IPOCL utilizes two additional types of flaws: *open motivation flaws* and *intent flaws*.

Definition 4.8 (Open motivation flaw). *An open motivation flaw in plan* P *is a tuple,* $\prec$c, p$\succ$, *such that* c *is a frame of commitment in* P *and* p *is the sentence* `(intends a g_a)` *such that* a *is the character of* c *and* g_a *is the internal character goal of* c.

Definition 4.9 (Intent flaw). *An intent flaw in plan* P *is a tuple,* $\prec$s, c$\succ$, *where* s $\in$ P *is a plan step and* c $\in$ P *is a frame of commitment such that* $s \xrightarrow{p} s_j$ *is a causal link in the plan,* s *is not part of* c, *and* $s_j \in$ P *is part of* c *and the character of* s *is the same as the character of* s_j *and* c.

Each of the new flaw types captures some aspect of fabula planning that makes it possible for IPOCL to search the space of plans in which character goals are possibly distinct from the planning problem goals and character goals are not declared before the plan begins. Open motivation flaws reflect the fact that characters must appear motivated to have goals. An open motivation flaw means that a plan has a frame of commitment whose interval of intentionality is not preceded by a motivating step. Intent flaws reflect the fact that a plan step, *s*, performed by a character can be part of the interval of intentionality of a frame of commitment, *c*, held by that same character. That is, step *s* causally establishes a precondition of some other step, s_j, which is part of *c*. One interpretation of the plan is that *s* is performed by the character to make s_j possible and that, since s_j is part of the interval of frame *c*, then *s* must also be performed as part of the character's plan to achieve the internal character goal of *c*.

Intention recognition simulation is applied to an incomplete plan node when a character action is newly instantiated or when a character action is reused. The purpose of intention recognition in IPOCL is to realize that a new character action may be intended as part of an existing frame of commitment or to recognize the character action as part of a new intention. The intention recognition process is simulated by IPOCL in the way that it opportunistically constructs new frames of intention and the way that it resolves open motivation flaws and intent flaws. The IPOCL algorithm, shown in Figure 4.4, is broken up into three parts: *causal planning*, *motivation planning*, and *intent planning*.

4.4.1. Causal Planning in IPOCL

The causal planning portion of the IPOCL algorithm implements the conventional POP algorithm with the addition of a frame of discovery phase. Causal planning occurs when there is an open condition flaw that needs to be resolved. That is, some step, s_{need}, has a precondition, p, that is not satisfied by any causal link. The planner chooses a plan step, s_{add}, whose effect, e, can unify with p. This is accomplished by non-deterministically choosing an existing plan step or by instantiating an action schema.

IPOCL ($\prec S, B, O, L, C\succ$, F, Λ)

I. **Termination**. If O or B is inconsistent, fail. If F is empty and $\forall s \in S, \exists c \in C \mid s$ is part of c, return $\prec S, B, O, L, C\succ$. Otherwise, if F is empty, fail.

II. **Plan Refinement**. Non-deterministically do one of the following.

- **Causal planning**
  1. **Goal selection**. Select an open condition flaw $f = \prec s_{need}, p\succ$ from F. Let $F' = F - \{f\}$.
  2. **Operator selection**. Let s_{add} be a step that adds an effect e that can be unified with p (to create s_{add}, non-deterministically choose a step s_{old} already in S or instantiate an action schema in Λ). If no such step exists, backtrack. Otherwise, let $S' = S \cup \{s_{add}\}$, $O' = O \cup \{s_{add} < s_{need}\}$, $B' = B \cup$ bindings needed to make s_{add} add e, including the bindings of s_{add} itself, and $L' = L \cup \{\prec s_{add}, e, p, s_{need}\succ\}$. If $s_{add} \neq s_{old}$, add new open condition flaws to F' for every precondition of s_{add}.
  3. **Frame discovery**. Let $C' = C$.
     a. If $s_{add} \neq s_{old}$, non-deterministically choose an effect e of s_{add} or e = nil. If $e \neq$ nil, construct a new frame of commitment, c with internal character goal e and the character of s_{add}, let s_{add} be part of c, let $C' = C \cup \{c\}$, create a new open motivation flaw $f = \prec c\succ$, and let $F' = F \cup \{f\}$.
     b. Let C'' be the set of existing frames of commitment that can be used to explain s_{add}. For all $d \in C''$, create an intent flaw $f = \prec s_{add}, d\succ$ and let $F' = F \cup \{f\}$.
  4. **Threat resolution**.
     - **Causal threat resolution**. Performed as in II.3 in the POCL algorithm (Figure 3.2).
     - **Intentional threat resolution**. For all $c_1 \in C'$ and $c_2 \in C'$, such that the character of c_1 is the same as the character of c_2, e_1 is the goal of c_1, and e_2 is the goal of c_2, if e_1 negates e_2, non-deterministically order c_1 before c_2 or vice versa and for all $s_1 \in c_1$ and all $s_2 \in c_2$, $O' = O' \cup \{s_1 < s_2\}$ or $O' = O' \cup \{s_2 < s_1\}$.
  5. **Recursive invocation**. Call IPOCL($\prec S', B', O', L', C'\succ$, F', Λ).

- **Motivation planning**
  1. **Goal selection**. Select an open motivation flaw $f = \prec c\succ$ from F. Let p be the condition of c. Let $F' = F - \{f\}$.
  2. **Operator selection**. Same as causal planning above, except $\forall s_i \in c$, $O' = O \cup \{s_{add} < s_i\}$.
  3. **Frame discovery**. Same as for causal planning, above.
  4. **Threat resolution**. Same as for causal planning, above.
  5. **Recursive invocation**. Call IPOCL($\prec S', B', O', L', C'\succ$, F', Λ).

- **Intent planning**
  1. **Goal selection**. Select an intent flaw $f = \prec s, c\succ$ from F. Let $F' = F - \{f\}$.
  2. **Frame selection**. Let $O' = O$. Non-deterministically choose to do one of the following.
     - Make s part of c. Let s_m be the motivating step of c. $O' = O' \cup \{s_m < s\}$. For all $c_i \in C$ such that c_i is ordered with respect to c, then for all $s_i \in c_i$, $O' = O' \cup \{s_i < s\}$ or $O' = O' \cup \{s < s_i\}$. For each $s_{pred} \in S$ such that $\prec s_{pred}, p, q, s\succ \in L$ and s_{pred} and s have the same character, create an intent flaw $f = \prec s_{pred}, c\succ$ and let $F' = F' \cup \{f\}$.
     - Do not make s part of c.
  3. **Recursive invocation**. Call IPOCL($\prec S, B, O', L, C\succ$, F', Λ).

Figure 4.4. The IPOCL algorithm.

4.4.1.1. Frame Discovery

The simulated intention recognition process, as implemented by frame of commitment discovery, is triggered by the changes in the plan (e.g. the addition of a causal link to the plan structure). If s_{add} is a newly instantiated action, then there is the possibility that it is the final step (due to the backward-chaining nature of the planning algorithm) of some previously undiscovered character intention. If this is the case, then one of the effects of s_{add}, in addition to causally satisfying some open condition, is intended by the character performing s_{add}. IPOCL non-deterministically chooses one of the effects of s_{add} (or no effect, in the case where s_{add} is not the final step of some undiscovered intention). If an effect is chosen, then a new frame of commitment is constructed to record the character's commitment to achieving that effect in the world. Step s_{add} is made to be the final step of the frame's interval of intentionality and a new open motivation flaw annotates the plan to indicate that the planner must find a plan in which the character's commitment to the new goal is motivated.

Table 4.1. Two action schemata demonstrating joint versus singular character intentionality.

```
(define (action duo-car-ride)
   :parameters (?driver ?rider ?car ?start ?destination)
   :actors (?driver ?rider)
   :constraints ((person ?driver) (person ?rider) (car ?car)
                 (place ?start) (place ?destination))
   :precondition ((in ?driver ?car) (in ?rider ?car)
                  (driving ?driver ?car)
                  (at ?car ?start) (at ?driver ?start) (at ?rider start))
   :effect ((at ?driver ?destination) (at ?rider ?destination)
            (at ?car ?destination)))
```

```
(define (action duo-car-ride)
   :parameters (?driver ?rider ?car ?start ?destination)
   :actors (?driver)
   :constraints ((person ?driver) (person ?rider) (car ?car)
                 (place ?start) (place ?destination))
   :precondition ((in ?driver ?car) (in ?rider ?car)
                  (driving ?driver ?car)
                  (at ?car ?start) (at ?driver ?start) (at ?rider start))
   :effect ((at ?driver ?destination) (at ?rider ?destination)
            (at ?car ?destination)))
```

Until now, the discussion has assumed that an action is performed by a single character. An action schema has a certain number of parameters, any of which can reference a story world character. The previous assumption was that one of these parameters referred to a story world character that was intentionally acting and all other references to story world characters implied characters that were being acted upon. However, there is a class of actions in which more than one story world character are acting intentionally, such as `Duo-Car-Ride(?driver, ?rider, ?car, ?start, ?destination)`, in which two people consensually ride in a car with the intention of getting from one place to another. Two version of the `Duo-Car-Ride` schema are shown in Table 4.1 (the differences are highlighted). The additional slot, `:actors`, specifies which of the parameters are characters

Table 4.2. Possible frames of commitment generated for action schemata with joint and singular character intentionality.

Frames of commitment for action schema with joint character intentionality	Frames of commitment for action schema with singular character intentionality
o ?driver intends (at ?driver ?destination) o ?rider intends (at ?driver ?destination)	
o ?driver intends (at ?driver ?destination) o ?rider intends (at ?car ?destination)	
o ?driver intends (at ?driver ?destination) o ?rider intends (at ?rider ?destination)	
o ?driver intends (at ?driver ?destination) o ?rider does not intend any effect	o ?driver intends (at ?driver ?destination)
o ?driver intends (at ?car ?destination) o ?rider intends (at ?driver ?destination)	
o ?driver intends (at ?car ?destination) o ?rider intends (at ?car ?destination)	
o ?driver intends (at ?car ?destination) o ?rider intends (at ?rider ?destination)	
o ?driver intends (at ?car ?destination) o ?rider does not intend any effect	o ?driver intends (at ?car ?destination)
o ?driver intends (at ?rider ?destination) o ?rider intends (at ?driver ?destination)	
o ?driver intends (at ?rider ?destination) o ?rider intends (at ?car ?destination)	
o ?driver intends (at ?rider ?destination) o ?rider intends (at ?rider ?destination)	
o ?driver intends (at ?rider ?destination) o ?rider does not intend any effect	o ?driver intends (at ?rider ?destination)
o ?driver does not intend any effect o ?rider intends (at ?driver ?destination)	
o ?driver does not intend any effect o ?rider intends (at ?car ?destination)	
o ?driver does not intend any effect o ?rider intends (at ?rider ?destination)	
o ?driver does not intend any effect o ?rider does not intend any effect	o ?driver does not intend any effect

that can be acting intentionally. Note that the first schema specifies that both characters are acting intentionally while the second schema specifies that the driver is acting intentionally and the rider is not, perhaps implying that the rider is being taken somewhere that he does not intend to go. Table 4.2 shows the possible frames of commitment that can be created when the `Duo-Car-Ride` action schema is instantiated as a step in a plan (both the joint character intentionality version and the singular character intentionality version). For one unique binding of the action's five parameters, each row in the table signifies a branch in the search space if there are only two characters, one car, and two locations in the story world.

Joint intention actions are either rare or common, depending on the relative level of abstraction at which the plan library is engineered. For example, `Duo-Car-Ride` is at a relatively high level of abstraction compared to the various physical, individual behaviors that make up the act at a more primitive level. Support for joint intention actions does not significantly change the IPOCL algorithm. When a joint intention action is instantiated from an action schema, the planner must non-deterministically consider all possible combinations of intentions that the character can have and construct a frame of commitment for each character involved. No frame of commitment for any character involved still remains an option.

When an open condition flaw is repaired by instantiating an action schema, there is a branch in the search space for every possible combination of parameter bindings and frames of commitment (including no frame of commitment) for each character that could be participating in the action intentionally. As a result, the branching factor can be quite large. The branching factor is further increased when more than one character is participating intentionally in the execution of an action. One way to limit the branching factor is to distinguish between the effects of an action that can be intended and the effects of an action that are never intended. The effects of an action that are not used to establish preconditions of future actions are *side effects*. The effects of an action that are not used as internal character goals are *unintended effects*. The effects of an action that *cannot* be used as internal character goals are *unintendable effects*. When specifying an action schema the domain engineer can observe that certain effects are never intended. For example, if the story is set during a war and there is a school house next to a munitions depot, a bomb dropped from an airplane onto the depot will destroy both the depot and the school house. A bomber pilot sent to destroy the munitions depot intends to destroy the depot but does not intend to destroy the school house, even though its destruction is one of the effects of dropping a bomb; the effect of destroying the school house is a side effect of dropping a bomb (Bratman, 1990). The assumption that the bomber pilot will never intend to destroy a school house makes the side effect an unintendable effect. The pilot may intend to bomb the munitions depot and, in doing so, destroy the school house, but the story planner will not conceive of a story plan in which the pilot intends to destroy the school house. Since an unintendable effect can never be intended, IPOCL can immediately prune any branch in which a character has a frame of commitment with a side effect for an internal character goal. Denoting certain effects on an action schema as side effects effectively limits the story planner from considering stories where a character intends to achieve those conditions in the world. The usage of side effects is optional; declaring side effects trades efficiency for completeness since the branching factor is reduced at the expense of the range of world states a character can intend.

Regardless of whether s_{add} is newly instantiated or an existing plan step that is reused, the planner must consider the possibility that s_{add} is part of an existing interval of intentionality. This represents the fact that an action can be performed as part of more than one intention. This corresponds to the notion of *overloading* (Pollack, 1992). IPOCL performs a search of the plan node for frames of commitment that s_{add} can be part of. The search routine finds a set of frames, C'', such that if $c_i \in C''$, then one of the two following conditions holds.

- The frame of commitment, c_i, contains step s_j such that $s_{add} \xrightarrow{p} s_j$ is a causal link in the plan and s_{add} and s_j are performed by the same character.
- The frame of commitment, c_i, contains step s_j such that some frame $c_j \notin C''$ is in service of s_j and s_{add} is a motivating step for c_j.

For each frame of commitment $c_i \in C''$, the plan is annotated with an intent flaw, $\prec s_{add}, c_i \succ$. By resolving these flaws, the planner will determine whether step s_{add} becomes part of an existing frame's interval of intentionality.

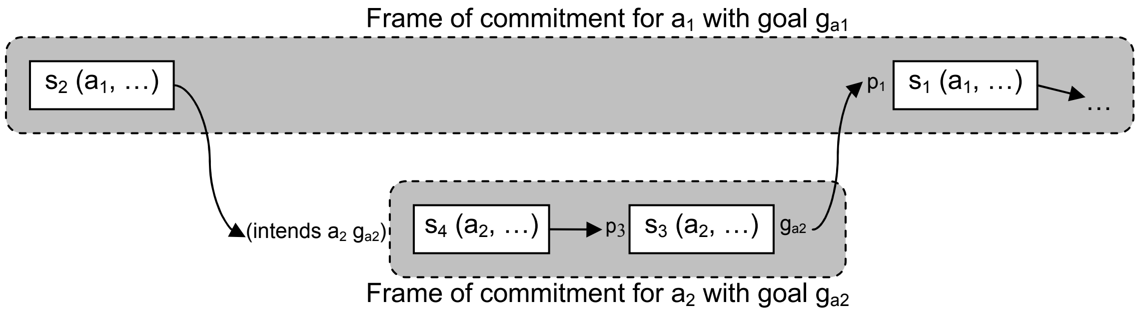

Figure 4.5. An IPOCL plan where one character is contracted out by another character.

Condition 1 indicates that if two actions, s_j and s_{add}, are performed by the same character and the earlier action, s_{add}, establishes some condition in the world required for the later action, s_j, then a reasonable hypothesis is that both were part of the same intention. The intent flaw on the earlier action indicates that the planner must, at some point, decide whether to support this hypothesis by incorporating the actions into the same interval of intentionality or to reject the hypothesis by leaving the plan structure unchanged. Condition 2 indicates the situation where an agent requires a certain world state to be achieved to make its intentional actions feasible and this sub-goal is contracted out to another agent (Grosz & Kraus, 1996). It occurs when a motivating action performed by one character causes another character to have an intention that is in service of the first character's actions, as demonstrated in Figure 4.5. Character a_1 performs an action, s_1, in pursuit of goal g_{a1}. Action s_1 has a single precondition that is satisfied by an action, s_3, performed by character a_2 in pursuit of goal g_{a2}. Action s_2 is the motivating action that causes character a_2 to have the goal to establish the precondition of step s_1. Since the motivating step is performed by character a_1, it is a candidate under Condition 2 to be incorporated into a_1's frame of commitment.

To remain consistent with the narratological distinction between acts and happenings (Prince, 1987), IPOCL considers any plan step that has a character who acts intentionally as an *act* and any plan step that does not have a character who acts intentionally as a *happening*. Action schemata in the action library can be declared as happenings, meaning that when they occur in the story, no character needs to intend that action to happen. Happenings can occur in two varieties: those that are performed by characters, and those that are not performed by characters. An example of a happening that is performed by a character is `Drop`, where a character unintentionally drops an object. An example of a happening that is not performed

Table 4.3. Examples of action schemata for happenings.

```
(define (action drop)
   :parameters (?char ?thing ?place)
   :actors (?char)
   :not-needs-intention t
   :constraints ((person ?char)
                 (thing ?thing)
                 (place ?place))
   :precondition ((at ?char ?place)
                  (has ?char ?thing))
   :effect (¬(has ?char ?thing)
            (at ?thing ?place)))
```

```
(define (action meteor-strike)
   :parameters (?met ?place)
   :actors nil
   :not-needs-intention t
   :constraints ((meteor ?met)
                 (place ?place))
   :precondition ((in-space ?met))
   :effect ((destroyed ?place)
            ¬(in-space ?met)))
```

by a character is `Meteor-Strike`, where a meteor crashes to the earth killing a character. The distinction is important because actions that need not be intended can be intended. For example, a character can intentionally drop an object whereas (presumably) no character can intend that a meteor fall from the sky. Actions in each sub-class are treated differently by the IPOCL algorithm. Actions that are happenings that are not performed by a character are ignored by the frame discovery portion of the IPOCL algorithm. Actions that are happenings that are performed by a character are treated as if it were a normal action by the frame discovery portion of the algorithm. The only difference here is that if the action is left an orphan, it is not counted against the completeness of the plan. Table 4.3 shows action schemata for `Drop` and `Meteor-Strike`. Note how schemata for happenings specify that characters – if any – do not need to be acting intentionally.

4.4.1.2. Threat Resolution

Once frame discovery takes place, the planner must resolve any threats that were inadvertently introduced into the refined plan. There are two types of threats, *causal threats* and *intentional threats*. A causal threat occurs when a new step, s_k, is instantiated that has an effect that negates the proposition in a causal link, $s_i \xrightarrow{p} s_j$. The standard POP algorithm corrects causal threats by non-deterministically ordering s_k before s_i or after s_j. IPOCL does not alter the process of causal link detection or correction.

In addition to causal threats, IPOCL must also deal with intentional threats, which occur when a newly instantiated frame of commitment, c_k, has an internal character goal that negates the internal character goal of some other frame of commitment, c_i, when both c_k and c_i share the same character. While it is possible for an agent – or character – to hold conflicting desires, it is not rational for an agent to concurrently commit to conflicting desires (Bratman, 1987). Character actions in the fabula plan may be unordered with respect to one another and this allows for intervals of intentionality that are interleaved. Interleaved intervals are not problematic except when the intervals belong to frames of commitment with conflicting goals. To prevent this, IPOCL detects intentional threats and corrects them by non-deterministically ordering c_i and c_j. The ordering of frames of commitment amounts to explicitly ordering the actions that are part of each frame to correspond to the ordering of c_i and c_j. When a new action is associated with one of these frames of commitment, it is ordered with respect to all the actions in the other frame of commitment.

4.4.2. Motivation Planning in IPOCL

The motivation planning portion of the IPOCL algorithm is responsible for ensuring that characters in the story world are motivated to have the intentions that they commit to. A motivating step is a plan step in which one of its effects causes a character to have an internal character goal. Repairing an open motivation flaw consists of non-deterministically finding a plan step with effect `(intends a` g_a`)` – either by choosing an existing plan step or by instantiating an action schema and explicitly ordering that step before the plan steps that are part of the frame of commitment's interval of intentionality. Motivation planning is similar to causal planning except instead of establishing a causal link between two plan steps, it establishes a motivation link between a motivating step and a frame of commitment. Additionally, the motivating step for a frame of commitment is explicitly ordered before all other steps in the frame's interval of intentionality. In the work presented here, an agent cannot begin pursuing a goal before the agent has committed to the goal, although it is typically possible and possibly even advantageous for an agent to adopt a strategy of performing actions that will facilitate achieving a goal should that agent actually commit to it at a later date. Motivation planning involves frame discovery and threat resolution phases that are identical to causal planning.

4.4.3. Intent Planning in IPOCL

The intent planning portion of the IPOCL algorithm determines interval membership for all character actions except those that are final steps for their intervals of intentionality. Intent planning repairs intent flaws. An intent flaw is a decision point that asks whether a plan step, *s* should be made part of the interval of some frame of commitment, *c*. Unlike other flaws that are repaired by refining the structure of the plan, intent flaws are resolved by non-deterministically choosing one of the following.

- Make step s part of the interval of c and refine the plan structure to reflect the association.
- Do not make step s part of the interval of c, remove the flaw annotation, and leave the plan structure unchanged.

When the former is chosen, step s becomes part of the interval of intentionality of frame c. When this choice is made, the interval of frame *c* is updated appropriately and *s* is explicitly ordered after the motivating step of frame *c*. Furthermore, the change in the step's membership status can have an effect on the membership of plan steps that precede *s*. Let s_{pred} be an establishing step of *s* – a step that precedes *s* and is causally linked to *s*. The inclusion of step *s* in the interval of frame *c* also makes it possible for establishing steps to be included in the interval of *c* if the following conditions hold.

- Step s_{pred} is performed by the same character as *s*.
- Step s_{pred} is not a part of the interval of intentionality of *c*.
- The intent flaw, $f = \prec s_{pred}, c \succ$ has not already been proposed and/or resolved[6].

The plan is annotated with intent flaws for each establishing step for which all three conditions hold. Intent planning thus operates in a spreading activation fashion. When one step becomes a member of a frame of commitment, an entire sequence of establishing steps

[6] The inclusion of this condition ensures the systematicity of the algorithm since there can be more than one causal link between s_{pred} and *s*. A search algorithm is systematic if it is guaranteed to never duplicate a portion of the search space.

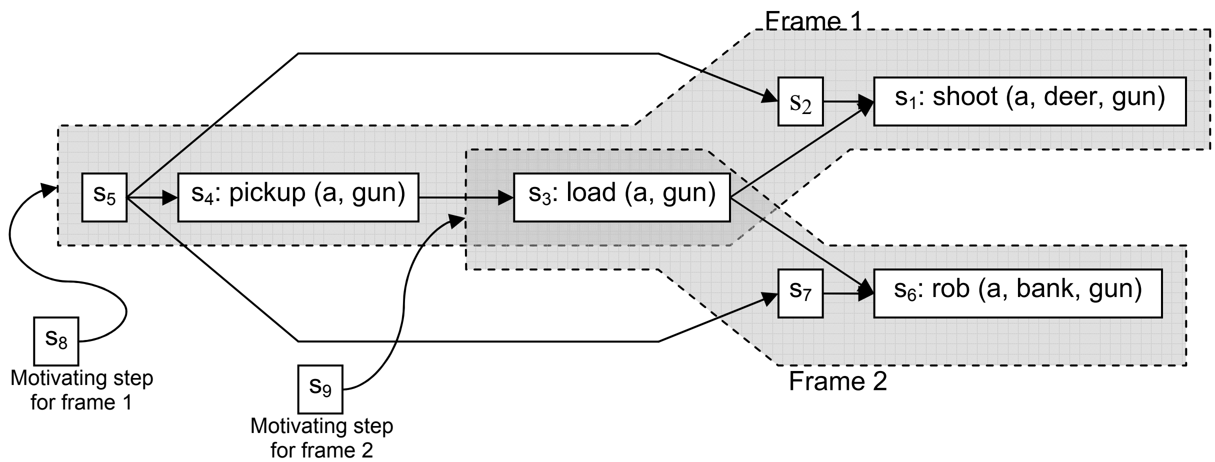

Figure 4.6. An IPOCL plan with two frames of commitment sharing steps.

may follow. This approach is necessary since frames of commitment can be created at any time during plan refinement.

4.4.3.1. Supporting Actions for Multiple Commitments

The propagation of intent flaws makes it possible for plan steps to become members of more than one frame of commitment, which is a desirable property of the IPOCL algorithm. Every time a character action – belonging to one frame of commitment – is used to satisfy an open condition of a successor action that belongs to a different frame of commitment, the system must non-deterministically decide whether the establishing action belongs to both frames of commitment or remains only a member of its original frame. The decision about interval membership also constraints the possible ordering of motivating steps for the frames of commitment involved. Motivating steps are temporally ordered before all actions in the interval of the frame of commitment that the motivating step establishes. Any motivating step can be placed temporally in the plan at any point before the interval of its frame of commitment begins. When an action becomes a member of more than one frame of commitment, the possible placement of motivating steps is constrained as in Figure 4.6.

For example, if a character has an internal goal of killing a deer as well as an internal goal of robbing a bank, then that character may pick up and load a gun as part of his commitment to one or both of those internal goals. If the act of loading the gun is part of both intentions, then the character must form intentions to kill a deer and rob the bank before loading the gun.

That is, motivating steps for both intentions must be ordered before the loading operation. However, if loading the gun is only part of the intention to kill a deer, then the character can form the intention of robbing the bank after the gun is loaded.

The ordering of motivating steps can lead to ambiguity of audience interpretation. For example, suppose the planner specifies the loading the gun only to be part of the character's intention to kill a deer. The motivating step may still be ordered before the load action. While the planner has represented a very specific relationship between the two goals and the load action, the audience is free to interpret the loading of the gun is part of the character's intention to rob the bank. Algorithmically, nothing can be done to prevent this misinterpretation by the audience due to the partially ordered nature of POP plans. However, additional ordering constraints that are consistent with the ordering imposed by the IPOCL algorithm (possibly even forcing a total ordering) can be added to the fabula plan by a post-production process (see Section 7.1.2) and heuristics are provided that favor orderings that place motivating steps as close to the beginning of their corresponding intervals of intentionality as possible.

Figure 4.6 illustrates two overlapping frames of commitment for a single character, *a*. Suppose that the steps in the plan shown are added as numbered. By solving intent flaws, the interval of intentionality of Frame 1 spreads to incorporate steps s_1 through s_5. Similarly, the interval of intentionality of Frame 2 initially spreads to incorporate s_6 and s_7. Given that step s_3 is an establishing step for actions in Frame 2, the planner has two choices: s_3 can remain solely in Frame 1 or it can assume joint membership with Frame 2. If the latter option is selected, the interval membership status of s_4 comes into question. The planner decides whether s_4 remains part of the first frame or whether it assumes joint membership as well. Had s_3 been originally left an orphan, the planner would have had different choices to consider: s_3 could remain an orphan or s_3 could join the interval of Frame 2. Thus, through two decision points, all four membership possibilities exist for step s_3: orphan, member of Frame 1, member of Frame 2, or member of both frames.

4.4.3.2. Orphans

IPOCL allows actions to be orphans (i.e. to not belong to any frame of commitment) if the plan is an internal node in the fabula plan search space. Orphans are allowed in order to avoid making an overly strong commitment to the interval membership of an action. For completeness, an action might need to be part of a frame of commitment that has not been discovered yet. Orphaned actions represent flaws in the plan because a plan is not complete unless all actions are intentional or are happenings. Unlike other flaws such as open conditions and open motivations, orphaned actions are not explicitly repaired. Instead, orphans are surreptitiously repaired when they are adopted into new intervals of intentionality because they causally establish other, intentional actions. Orphaned actions cannot be repaired directly because frames of commitment are discovered opportunistically instead of instantiated in a least-commitment approach (as plan steps are).

The strategy of leaving orphans with the hope that they will be adopted eventually is not without some risk. It is possible that an orphan is never used to establish another open condition. In this case, the orphan will never be adopted and the plan can be causally complete but not complete with respect to intentionality. In this occurs, the leaf plan is simply pruned and the planner backtracks to find another solution plan.

4.5. An Example

The IPOCL algorithm is illustrated by the following story about an arch-villain who bribes the President of the United States with a large sum of money. The example traces a single path through the fabula plan search space generated by IPOCL. The initial plan node contains only the initial state step and goal state step. The initial state contains sentences describing the state of the world before the story begins. The goal state contains a single sentence, `(corrupt President)`, which describes what must be different about the world after the story is complete. The story that will be generated by IPOCL is, in effect, the story about how the President becomes corrupt.

The goal sentence, as an open condition is non-deterministically established by instantiating a new character action, `Bribe(Villain, President, $)`, which states that the Villain character will bribe the President character with some money. The `Bribe` action was chosen because it has `(corrupt President)` as an effect. From the planner's perspective, the `Bribe` action is causally motivated by the open condition of the goal state. The audience, as active problem-solvers, is free to consider different reasons for this action to take place. Specifically, the audience will assume that the Villain bribes the President because of some commitment the Villain has to some internal character goal. Upon instantiation of the `Bribe` action, intention recognition is invoked in the form of frame discovery. Suppose the effects of the Bribe action are as follows.

- `(corrupt President)` – the President is corrupt.
- `(controls Villain President)` – the Villain exerts control over the President.
- `(has President $)` – the President has the money.
- `¬(has Villain $)` – the Villain does not have the money.

From the audience's perspective, any of these effects can be a reason why the Villain performs the actions in the story. The planner non-deterministically chooses `(controls Villain President)` as the internal character goal for the Villain character. Note that in this case the goal of the Villain differs from the outcome of the story although the same action satisfies both conditions. There is no reason why the planner could not have chosen `(corrupt President)` as the internal character goal for the Villain. It is assumed here that either the plan cannot be completed if the alternative is chosen or that some heuristic function has evaluated all options and determined that villains are more likely to want control over the President than anything else. Given the choice made, the planner constructs a frame of commitment for the Villain character and makes the `Bribe` action the final step in the frame's interval of intentionality. Even with the new frame of commitment, the plan is still flawed since there is no reason for the Villain character to have the internal goal of controlling the President. That is, the Villain needs to *form* the intention to appear believable to the audience. An open motivation flaw annotates the plan, indicating that some action in the plan must satisfy the condition `(intends Villain (controls Villain President))` on the frame of commitment.

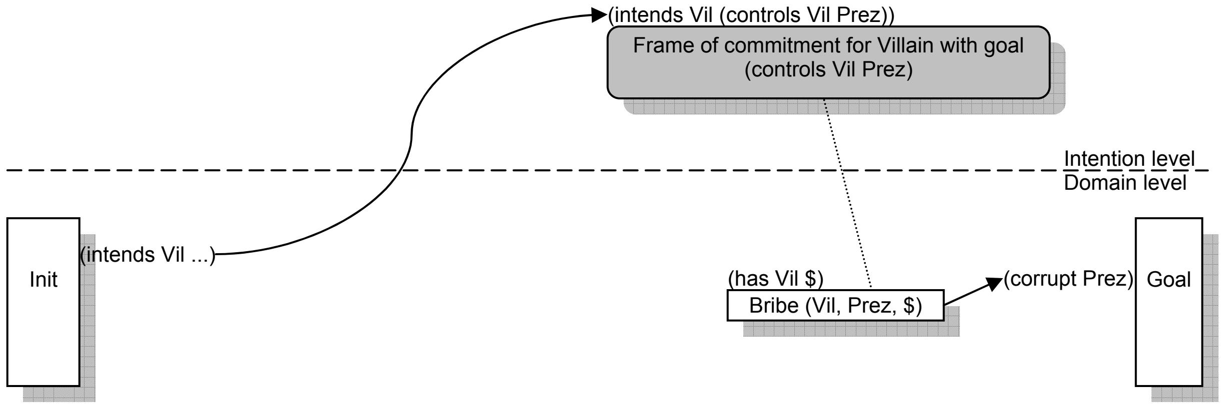

Figure 4.7. A partial IPOCL plan with a single frame of commitment.

Since there are no other frames of commitment for the Villain, no intent flaws occur. The `Bribe` action, however, has a precondition, `(has Villain $)`, which becomes an open condition; the Villain character must have the money if he is to bribe the President with it. However, the planner chooses to repair the open motivation flaw on the single frame of commitment first (the order in which flaws are resolved does not affect the completeness of the algorithm). The planner non-deterministically chooses the initial state to satisfy the open motivation condition. This illustrates a situation where the intention of a character in the story world is encoded as part of the initial conditions. While it does not have to be this way, the domain engineer that specified the inputs to IPOCL has decided that no further motivation for the Villain to want to control the President is needed. While this may not be the most satisfactory solution, it is a valid solution. The partial plan at this point is shown in Figure 4.7.

The open condition, `(has Villain $)`, on the `Bribe` action is considered next. To repair this flaw, the planner non-deterministically instantiates a new character action, `Give(Hero, Villain, $)`, in which the Hero character gives the Villain the money. The planner must consider, from the audiences, perspective, why the Hero character gives the money to the Villain. The planner inspects the effects of the `Give` action, which are given as follows.

- `(has Villain $)` – the Villain has the money.
- `¬(has Hero $)` – the Hero does not have the money.

The planner non-deterministically chooses `(has Villain $)` as the goal that the Hero is attempting to achieve. A new frame of commitment for the Hero's goal is created. Note that the Hero's intention matches the open condition that the `Give` action was instantiated to satisfy. This indicates that the Hero's commitment is in service to the `Bribe` action.

An open motivation flaw is created that corresponds to the new frame of commitment. There are many actions that will establish the Hero's intention that the Villain has the money: the Villain might persuade the Hero if they are friends, or the Villain might coerce the Hero. The latter, `Coerce(Villain, Hero, (has Villain $))`, is chosen by the planner. Syntactically, the Villain character coerces the Hero character into having the goal, `(has Villain $)`.

At this point, the planner must determine why the Villain coerces the Hero. There are several possibilities. First frame discovery comes into play to determine if the Villain intends any of the effects of the `Coerce` action. Assume the only effect of the `Coerce` action is `(intends Hero (has Villain $))`. The planner can select this effect and construct a new frame of commitment specifying that the Villain intends that the Hero intends that the Villain has the money. Another option is to leave the `Coerce` action an orphan for the time being. Let us suppose that this is the course that the planner chooses. A search of the current plan structure indicates that the `Coerce` action can be part of the Villain's existing commitment to control the President. This is possible because `Coerce` is a motivating step for the Hero's frame of commitment and the Hero's frame of commitment is in service to the `Bribe` action, which is part of the Villain's frame of commitment. An intent flaw associating the `Coerce` action with the Villain's existing frame of commitment is created.

The planner next repairs the intent flaw on the `Coerce` action. The planner can choose to associate the action with the frame of commitment or choose to leave the plan structure unmodified. The planner non-deterministically chooses to make the `Coerce` action part of the Villain's intention to control the President. If the planner had chosen otherwise, `Coerce` would remain an orphan and any causally complete solution plan that was found at that point will be rejected by IPOCL because a step that is not intentional remains. The plan structure at this point is shown in Figure 4.8.

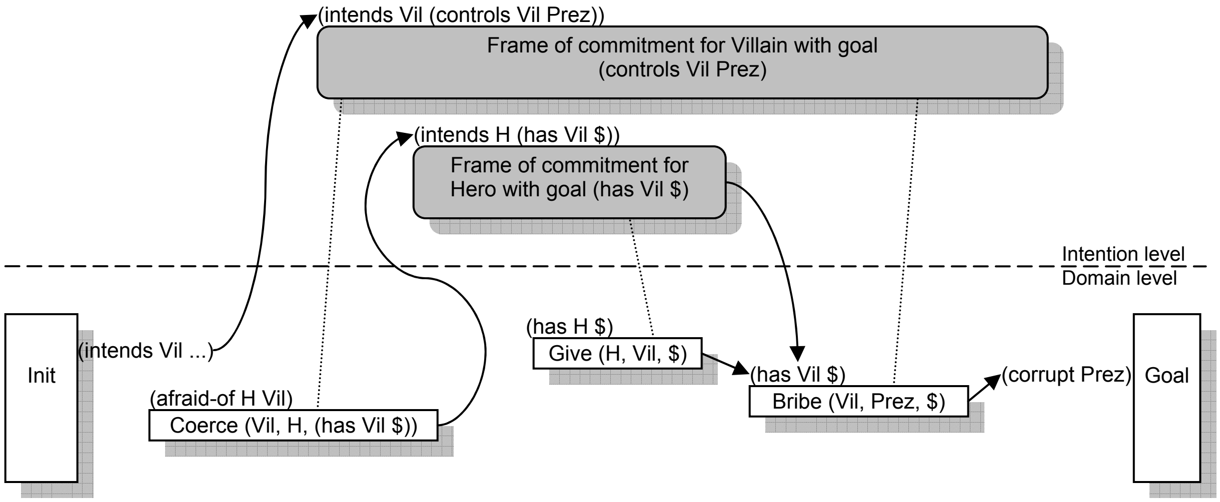

Figure 4.8. A partial IPOCL plan with several characters with interrelated frames of commitment.

There are two flaws remaining in the plan. The `Give` action has an open condition, `(has Hero $)`, indicating that the Hero must have the money if he is to give it to the Villain. The `Coerce` action has an open condition, `(afraid-of Hero Villain)`, indicating that coercion is only possible if the Hero is afraid of the Villain. As the planner continues to refine the plan, the open condition, `(has Hero $)`, is established by a new character action, `Steal(Hero, $, bank)`, in which the Hero steals the money from the bank. The `Steal` action establishes a precondition of the `Give` action and can therefore be adopted into the Hero's frame of commitment as long as `Steal` is ordered to occur after the `Coerce` action – the motivating step of the Hero's frame of commitment. Likewise, the open condition, `(afraid-of Hero Villain)`, is established by a new character action, `Threaten(Villain, Hero)`, in which the Villain threatens to harm the Hero. The Threaten action establishes a precondition of the `Coerce` action and can therefore be adopted into the Villain's frame of commitment.

4.6. Heuristics for Fabula Planning

IPOCL is a domain independent fabula planner, meaning that it makes no assumptions about the way in which the story world is represented by world state sentences. It is beneficial, however, to inform the planner of domain-specific information so the planner can focus its search on branches of the search tree that are more likely to produce satisfying solutions.

Heuristic functions rank the plans on the fringe of the developing search tree according to some domain-specific definition. Typically, the heuristic is devised to guide the planner to the first complete solution plan, so as to shorten computation time. As with any type of planning, some solutions are better than other solutions. Particular to fabula planning is the notion that any solution (usually the first) is not necessarily acceptable.

The IPOCL planning algorithm opens up a larger search space in which character goals are possibly distinct from the outcome of the story. Increasing the search space places more emphasis on effective heuristic functions to focus the search on promising branches of the search tree. Due to the great variety in storytelling techniques and story genres, it is impossible to identify any metric for ranking a story plan that is common across all story world domains. Common heuristics for planning include factors such as plan length and number of flaws. Others that can be considered include number of frames of commitment, and size of intervals of intentionality. For most types of stories, it is probably not desirable to minimize the length of plan in terms of plan steps since characters – based on humans – do not always execute optimal plans. However, this metric can be useful for avoiding local minima. A heuristic for short stories would require a metric for plan length. The number of frames of commitment and the size of intervals of intentionality may also vary between genres. It is possible for the IPOCL planner to generate a new frame of commitment for every step that is instantiated in the plan. While this is legitimate, it would result in a proliferation of character intentions with intervals of intentionality containing only one action. Fiction oriented towards adults might involve deliberative characters, indicating fewer frames of commitments and longer intervals of intentionality. A proliferation of frames of commitment with small intervals of intentionality might be acceptable for children's stories.

Other genre-specific heuristics may include qualitative assessments of plans, such as the degree of interleaving between intervals of intentionality of different characters and the relative position of certain character actions in the timeline (e.g. a murder mystery should have the murder take place near the beginning of the story as opposed to later on). Heuristic metrics may also quantitatively measure the appropriateness of the story world state. Story world specific metrics include qualitative assessments about locations in which actions occur and the likelihood of certain actions occurring at certain times.

In general, writing heuristics that define the "goodness" of a story is a challenging prospect. No computational definition of story "goodness" exists on which to base computational functions that rank fabula plans. Writing heuristics that define the "goodness" of a fabula plan is further complicated by the fact that heuristics are applied to the fringe of a continuously filled out search space, meaning that a heuristic must evaluate fabula plans that are incomplete. The factors listed above are not comprehensive, nor is there a clear process of determining which metrics are appropriate for any given genre or style of storytelling. To date, only a few simple heuristics that measure plan length, number of flaws, and number of frames of commitment have been implemented.

4.7. Computational Complexity of the IPOCL Algorithm

Construction of plans with actions whose effects are deterministic is NP-Complete (Chapman, 1987). As reported by Weld (1994), the expected performance of a the POP algorithm is $O(cb^n)$ where

- n is the number of non-deterministic choices that must be made before a solution is obtained,
- b is the number of possibilities that need to be considered for each non-deterministic choice, and
- c is the time it takes to process a given node in the search space.

The parameter n corresponds to the depth of the search space and the parameter b corresponds to the branching factor. The cost per node factor c is often disregarded since it is dominated by b^n.

IPOCL expands the structural representation of a partially ordered plan in order to find plans in which the outcome of the story is possibly not intended by any agent and in which agents are not necessarily cooperating. In doing so, IPOCL opens up a larger search space.

Although IPOCL is also NP-Complete, it is interesting to compare the size of the search space for IPOCL problems to the search space for POP problems. The most significant difference between the complexity of IPOCL and the complexity of POP is the branching factor. In POP, the worst-case branching factor occurs during open condition refinement where any number of action schemata and any number of existing plan steps can be used to achieve a single open condition. In IPOCL, the worst case branching factor must consider the same existing plan steps and the same action schema. However, when considering instantiating a new plan step, IPOCL must also consider whether it is a final step of a newly discovered frame of commitment and, if so, which of the effects of that new plan step might be the internal character goal. The branching factor of the IPOCL search space is computed using the following two additional parameters.

- *e*, the largest number of effect propositions of all the action schemata in the action library.
- *a*, the largest number of characters that participate in a single action.

The worst-case branching factor of the IPOCL search space is $b(e+1)^a$, compared to the worst-case branching factor of the POP search space, which is *b*. The IPOCL algorithm, in the worst case, multiplies the branching factor by $(e+1)^a$. The factor, $(e+1)$ signifies that if a new frame of commitment is being constructed, the planner must choose between the *e* effects of the action, plus one to signify the condition where no effect is chosen. The exponent, *a*, reflects the fact that if multiple characters are intentionally participating in an action, then each of those characters can have distinct intentions for performing that action.

The computational complexity of the IPOCL algorithm therefore is $\mathrm{O}\left(c\left(b(e+1)^a\right)^n\right)$. This is misleading in that the IPOCL algorithm makes more non-deterministic choices than would the POP algorithm on a similar planning problem. The depth of the POP search space, *n*, is the number of open condition flaws and causal threats that are repaired. The depth of the IPOCL search space is the number of open condition flaws, open motivation flaws, intent flaws, causal threats, and intentional threats that are repaired. In the worst case scenario, for every newly instantiated step in the plan, IPOCL also creates a new frame of commitment.

With every frame of commitment comes an open motivation flaw. If n_{POP} is the depth of the search space of POP planning problem and n_{IPOCL} is the depth of the search space of a similar IPOCL planning problem, then n_{IPOCL} is bounded by the function, $n_{IPOCL} = 2n_{POP}$ in the worst case (where every plan step is a single step in an interval of intentionality), indicating that the depth of the IPOCL algorithm's search space can be up to twice as deep as the POP algorithm's search space. Practical experience with the IPOCL algorithm shows that increased branching factor of the search space is the most significant aspect of the increased complexity. In the worst case, IPOCL generates a significantly larger number of children nodes for any given plan node and many of these are very similar – differing only by the internal character goal in a frame of commitment – making it difficult to write heuristics that distinguish between siblings[7].

4.8. Summary

Partial-order planners are limited in their ability to generate stories because of assumptions about the way that planners are used. The goal of the planning problem is assumed to be a partial description of the world that some agent intends to bring about. Since the goal is intended, the agent's actions in the solution plan itself are assumed to be intentional. Consequently, a story planner is limited in its ability to generate stories that have strong character believability because the planner will only insert character actions into the plan that achieve the goal state. However, a story plan might have more than one agent/character. If the characters are to be believable, a story planner must consider the possibility that none of the characters intend the goal – the outcome – of the story. Furthermore, the actions that characters perform in the story world should appear to be intentional. That is, the audience should be able to perceive that the characters have committed to distinct goals and that the actions they perform are in direct pursuit of those goals. The IPOCL planning algorithm is based on the standard POCL algorithm and meets all of the requirements for intentional character action.

[7] The problem of defining heuristic functions that distinguish between sibling nodes in the plan search space arises in all POP algorithms, but is exacerbated in IPOCL due to the increased number of structural features that need to be distinguished.

The IPOCL algorithm uses a process not dissimilar from intention recognition. When actions are instantiated in the plan, the planner determines a plausible internal goal for the character to have that explains why the character would perform that action. A frame of commitment data structure is created for every internal goal that is found for every character. The creation of a frame of commitment causes an open motivation flaw to be created which can only be repaired by inserting an action into the story plan that causes that character to commit to the internal goal. Only when all character actions (that are not happenings) are parts of the interval of intentionality of some frame of commitment, can an IPOCL plan structure be considered complete. The extension of the planning algorithm and the plan representation comes at a cost: the computational complexity of the IPOCL algorithm is greater than that of the standard POCL algorithm. The IPOCL algorithm, however, is still sound. Although plausible explanations for the actions that story world characters perform are explicitly represented in the plan structure, there is no guarantee that the story audience will perceive the motivations and goals of the characters exactly as the plan specifies. This is due to the fact that the audience must draw its own conclusions about character motivations from the events in the story that they witness.

Chapter 5

Domain-Independent Support for a Personality Model

Character intentionality is only one part of character believability. Under the model of character believability, provided in Section 3.2.2, for a character to be considered believable, that character must also act consistently. By consistent, I mean that the behaviors performed by a character suggest a tendency to act regularly and predictably over time. Lebowitz (1984) defines character consistency as not having any internal contradictions. For Lebowitz, maintaining consistency means avoiding unexpected, observable changes in a character's description or history. For example, a 34 year old character should not become 47 overnight nor should bitter enemies become close friends from one moment to the next. Ensuring that a character act consistently, however, involves more than avoiding contradictions from one moment to the next; consistency of a character must be perceived by the audience. For an audience to perceive a character as acting consistently, a pattern of desires and actions must be observable over time. Determining whether a character is acting consistently is difficult, especially when the audience's exposure to that character is limited. For an audience to determine whether a character is acting consistently, members must be able to recognize trends in the history of the characters actions. Psychologists suggest that the audience actively models the characters in a story (Gerrig, 1993). That is, the audience forms hypotheses about the traits that a character possesses and evaluates future actions the character makes, reinforcing or rejecting hypotheses. Over time, a model is formed that helps the audience predict the way in which a character will attempt to achieve its goals. Despite the assumption that rational characters will act consistently, the process of model formation employed by the audience is complicated by the fact that characters do not *have* to act consistently.

The model of a character formed by the audience is not necessarily the same as the model of a character possessed by the author. It is the responsibility of the author to portray the character and consistency of behavior is one way in which the author can make the character's internal traits clear to an audience (Egri, 1960). It is therefore reasonable that a fabula planner contains models of story world characters it can use to build plans containing appropriate character behaviors. Consistency of character behavior is therefore defined relative to the character model that author and audience share. This type of model is used by the fabula planner to determine whether characters are acting consistently by comparing the actions generated during plan construction to the character models. In this chapter, I present a framework for incorporating a trait-based model of character psychology into a fabula planner in order to inform the non-deterministic process of action selection such that character consistency can be achieved.

5.1. Psychology of Personality

Personality is considered by psychologists to be the individual differences of people that account for their consistent patterns of behavior (Pervin, 1993). There are many different theories that account for personality with little or no commonality. While each branch of psychology has put forth theories of personality that fit with their larger interpretation of human thought and behavior, trait theories in particular have remained popular. Traits are distinguishing qualities of a character or person. Trait theories of personality assume that individuals differ along certain dimensions that correspond to traits. How much individuals differ is a matter of measuring the amount or quantity of each dimension. Unlike other fields of psychology, trait theory research involves the mathematical process of factor analysis instead of relying on any one theory of thought or behavior. A typical factor analysis approach to determining a set of personality traits is to take a vocabulary of descriptive words and group them into clusters. The clusters of words are given descriptive terms and these terms define the set of personality traits. Traits are then correlated with behaviors. As an example, one trait theory that has consistently withstood the rigors of empirical evaluation is the Five Factor Model (FFM) by Costa and McCrae (1985) which postulates five super-traits:

- Openness to experience (vs. closedness to experience)
- Conscientious (vs. Lack of conscientiousness)
- Extraversion (vs. Introversion)
- Agreeableness (vs. Disagreeableness)
- Neuroticism (vs. Emotional stability)

By defining the opposite of each trait, a continuum is created. An individual's personality can be summed up as data points along each dimension. Each super-trait is broken into subordinate traits such as impulsiveness, warmth, altruism, and dutifulness (Costa & McCrae, 1985). The success of trait theories of personality is attributed to the fact that they rely upon everyday descriptors of personality (for example, the list compiled by Allport and Odbert (1936)) as the source of their factor analysis.

Trait theories are folk psychological models of human personality because they are based on the ways in which people naturally describe and summarize each other's differences. However, humans are complex organisms and the small set of adjectives one might use to describe the personality of another cannot be anything more than a generalization of one's consistent or persistent behaviors. Furthermore, even though a small set of trait adjectives is used to describe one's personality, one is not bound to act consistently all the time.

More recently, personality has been addressed by motivational psychology as the individual differences that cause one to have a tendency to adopt certain goals (Ford, 1992). From this perspective, personality is interpreted not as a set of traits, but as a set of general goals that persist so that a person is constantly motivated to act in a way that achieves those goals. A person is altruistic when she has the persistent goal that he acts in an altruistic fashion which must constantly be achieved.

5.2. Computational Personalities for Synthetic Actors

In the context of believable synthetic agents, personality is identified by Loyall (1997) as the most important aspect of character believability. In the work presented here, personality is one element that contributes to character consistency. The operational definition of character believability used here does not weigh intentionality versus consistency. Unlike work in rational agents, where the goal is to develop agents that are competent problem solvers, work in computational personality places emphasis on different, appropriate ways of achieving a goal where the effectiveness of an agent's goal-oriented behavior is not the essential factor (Reilly, 1996).

Reilly (1996) argues that the believability of an embodied, autonomous agent can only be achieved if an agent's computational personality affects all aspects of the agent's decision making. Instead of offering a computational model of personality in which values can be assigned to traits, Reilly provides a methodology for programming an agent such that its personality is expressive (Reilly, 1996), meaning that the behaviors performed by an agent clearly illustrates the personality traits of that agent. Each autonomous agent has its own pools of customized behaviors and natural language templates, implemented in a reactive planning language, HAP (Loyall, 1997), which Reilly extends to express emotion. The limitation of this approach is that agent behaviors are not modular and cannot be reused (Reilly, 1996).

A more modular approach for representing personalities is to use a computational model of personality in which character behaviors are selected from a pool of behaviors common to all agents. Candidate behaviors for a character at any given time are evaluated heuristically against some character-specific standard for appropriateness. The Virtual Theater Project (Hayes-Roth & van Gent, 1996) implements embodied, autonomous agents embodied that act out improvisational performances in real-time in a virtual world. The computational personality model implemented in the Virtual Theater Project uses a trait theory of personality (Rousseau & Hayes-Roth, 1998). The personality model defines three traits that address the specific design goals of the system: confidence, activity, and friendliness. An agent is initialized with qualitative values for each personality trait plus other factors that influence emotion and social attitude. Each action in the agent's repertoire has a personality profile that indicates an ideal value for each personality trait that the agent should have. When confronted with alternative actions that achieve the same goal, the agent chooses the action that its personality most closely matches.

Personality models have also been incorporated into story generations systems that use planning. The Universe program uses a trait-based personality model (Lebowitz, 1984). Each story world character is represented by a *person frame* which stores information about that character such as the character's name, stereotypes, traits, interpersonal relationships with other characters, and the character's history. Traits, such as intelligence, moodiness, and promiscuity whose values range in integral value from 0 to 10, are continuous dimensions and the degree to which a character manifests a trait is stored as an integer value. Traits such as intelligence and moodiness and promiscuity have ranges between 0 and 10. Traits such as guile, self-confidence, and niceness have ranges between -10 and 10 where a negative value indicates that that character has the opposite of the trait. Plot fragment schemata are selected by the planner in Universe based on whether they achieve a particular sub-goal in the story. In the situation where there is more than one plot fragment that is applicable to a particular sub-goal, the traits of characters involved in a candidate plot fragment help refine the possibilities. Plot fragment schemata are constrained so that only characters with certain trait values can participate in the plot fragment (Lebowitz, 1985). Constraints on plot fragment schemata force characters to act consistently in that they can only perform actions that have been declared consistent with their trait values by the system designers.

Instead of a trait-based approach, Rizzo et al. (1999) use a goal-based model of personality in a deliberative agent that builds and executes plans for a single agent. The planner generates behavior for an agent that is meant to help the user perform tasks and uses general goals such as belongingness, safety, and understanding. Although the planner uses a motivational theory of personality, the general goals that an agent has correspond to particular classes of personalities such as altruist, normative, selfish, spiteful, and suspicious; personality types are implied by the model although not explicitly represented. The agent is given a goal to achieve plus a set of general goals that define its personality. A plan is constructed that achieves the goals. The effects of plan operators, in addition to changing the state of the world by adding or deleting propositions about the world, also add or delete general goals. A set of control rules prune plans from the search space that use operators that delete the general goals that define the agent's personality.

5.3. A Framework for a Domain-Independent Personality Model

The goal-based model of personality used in (Rizzo et al., 1999) and the trait-based model of personality used in Universe (Lebowitz, 1984) are appealing because they are integrated into planning algorithms. There are two problems with the goal-based model in (Rizzo et al., 1999) that keep it from being a general solution. First of all, it conflates the distinction between planning problem goals and general goals. General goals are not solved for in the same manner that planning problem goals are solved for. Planning problem goals are solved for in a least-commitment approach by ensuring that there is some action in the plan that changes the world so that the goal is true. That is, there is a causal link from some plan step to the goal. General goals define the agent's personality but do not need to be directly satisfied by some step in the plan. Instead all steps in the plan must *not* have an effect that undoes a general goal. General goals are persistent in the sense that an agent always has the same general goals, but planning problem goals are also persistent in the fact that if some step undoes a goal in the world, another step will reestablish the goal. Chapman (1987) refers to these re-establishers as "white knights."

The second concern is that the planner prunes any plans from the search space in which an operator conflicts with the general goals. The goal-based model operates as if there is a causal link for each general goal running from the initial step to the goal step. Any operator that negates a general goal causes a causal threat that cannot be repaired (because that operator cannot be ordered before the initial step or after the goal step). Consequently, the planner cannot find a plan in which an agent acts contrary to its personality. Personality is *tendency* towards consistent behavior; it does not strictly constrain an individual. With regard to the full repertoire of behaviors that an agent can perform, the planner is incomplete (Rizzo et al., 1999).

The trait-based model used in Universe uses character traits strictly to constrain what plot fragments a character can participate in (Lebowitz, 1985). The use of personality traits to constrain character participation in plot fragments means that a character cannot act in a way that is contrary to his defined personality. As with the goal-based model discussed above, the strong constraints render the planner in Universe incomplete with regard to the full repertoire of behaviors that a character can perform.

The distinction between the trait-based model of personality in Universe and the goal-based model of personality in Rizzo et al. (1999) is subtle, especially since the general goals held by an agent suggest a stereotypical categorization of the agent's personality. The trait-based model defines an agent's personality as a set of descriptive terms and filters or ranks action alternatives based on similarities between the action and the descriptors. The goal-based model defines an agent's personality as a set of general goals that should be opportunistically achieved whenever possible and filters or ranks action alternatives based on whether an action achieves the general goals or is inconsistent with them.

The following describes a framework for integrating a domain-independent personality model into a fabula planner. In a fabula planner, there are many characters, each of which can have a distinct personality. The framework must be able to represent distinct personalities and influence the actions that are inserted into the plan for those characters to perform so that the characters appear to be acting consistently. To act consistently means for that character to overwhelmingly perform actions that are in agreement with its personality. However, a framework for personality based on a trait-based model of human personality relies on several simplifying assumptions. First of all, the framework for personality assumes that an agent's personality can be accurately represented by a relatively small set of trait adjectives. For an agent to accurately portray a human character, there are nuances of behavior that are not succinctly captured by the folk-psychological description of that agent.

This assumption further relies on the domain engineer's choice of trait adjectives. While I recommend the FFM model, the human author may find different sets of descriptors to be more relevant. Secondly, the framework for personality assumes that there is a direct correlation between the relatively small set of trait adjectives and the behaviors an agent actually perform in the world. While some actions are clearly representative of particular traits, many actions are neutral with regard to any specific personality characteristic. Furthermore, how one's personality is described is based on one's history of behaviors. That is, for one to be described as being extraverted, one does not have to perform overtly extraverted behaviors. Instead, one's behaviors over a long period of time should suggest a general tendency towards extravertedness. One's history of behavior could include both extraverted and introverted behaviors as long as the extraverted behaviors appear more salient.

Trait-based computational models of personality often constrain agent action in such a way that the agent cannot perform actions that are not consistent with the description of its personality. To preserve completeness of the planning algorithm with respect to the full repertoire of actions that can be performed in the world, the framework should allow characters to act inconsistently when necessary. The framework supports the use of an externally specified trait theory of personality and all characters in the story world are defined relative to the set of descriptive personality terms used by a personality model as specified by the domain engineer. The framework is domain-independent, meaning it does not commit to one particular set of traits. This independency is important since trait theories tend to be general. Furthermore, any folk psychological taxonomy of descriptive terms qualifies as a trait theory. The domain engineer is free to define her own taxonomy of personality traits.

5.3.1. Representation of Character Personality

The framework for integrating a domain-independent personality model into a fabula planner allows each character in the story world to have a distinct personality. Even though each character has a distinct personality, each personality is an instantiation of a single trait-based personality model. The personality model describes the traits that make up ones personality.

A specific instantiation declares where a single character's personality falls along the defined dimensions. For example, the personality model might use the traits of the FFM (see Section 5.1.). A specific character might be completely open to experiences, somewhat introverted, neither agreeable nor disagreeable, et cetera.

The framework requires the personality traits of all characters in the story world to be declared in the initial state of the world as binary atomic ground sentences about the characters. For example a character's personality might be represented in the initial state by the following sentences: (open a_1), (introverted a_1), ¬(agreeable a_1), ¬(disagreeable a_1), and so on. Note that by representing a character's traits as binary sentences, the framework necessarily restricts traits to being binary statements about a character. Either a character has a trait or does not have a trait. Relying on binary traits is somewhat limiting but simplifies the operator selection portion of the planning algorithm because it does not have to reason about the fractional degree to which a character expresses a trait. If the FFM theory is used for a personality model, a character can be neurotic or not neurotic. However, for a character to be defined as not neurotic does not mean that that character is necessarily emotionally stable. A character that is defined as emotionally stable implies that that character is also not neurotic. In a personality model with binary descriptors, there are set relationships between personality traits where otherwise there would be a continuum. The framework exploits these relationships to preserve the expressivity of the personality model.

The personality model specifies the binary traits that describe character personalities. It also designates polarity relationships between traits. For example, if greedy and philanthropic are trait descriptors, and the human author specifies them as polar opposite traits, then a character that is greedy is, by default, not philanthropic and vice versa. Polar opposite traits help establish dimensions that a character's personality can be quantitatively evaluated against. The binary nature of the traits makes a set interpretation more appropriate than a continuum interpretation. The continuum interpretation of polar opposite traits is shown in Figure 5.1 while the set interpretation of polar opposite traits is shown in Figure 5.2. Whether or not two trait descriptors are polar opposites is based on how the human author specifies the personality model. In this case, the human author believes greediness and

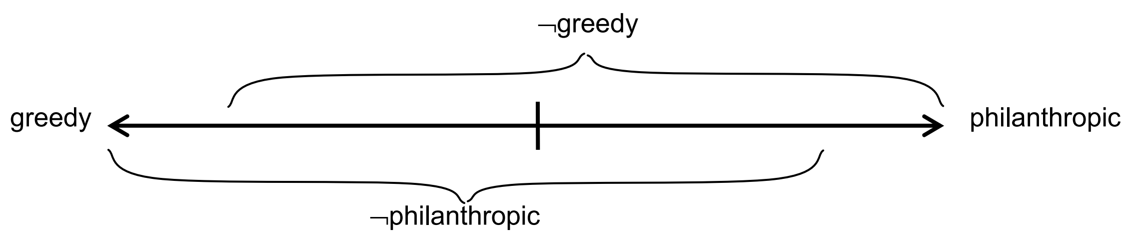

Figure 5.1. The continuum relationship of polar opposite personality traits.

philanthropy are opposites. Whether or not the audience sees the story world characters as behaving consistently or not depends on the degree to which they agree with the human author's representation of personality.

Most individuals do not express any one trait to an extreme degree. To fall anywhere in the middle of the continuum in Figure 5.1 would typically translate to being in the not-greedy and not-philanthropic intersection in Figure 5.2. But since some expressivity of the model is sacrificed based on simplifying assumption, I rely on the fact that effective story world characters – at least concerning the primary protagonists and antagonists – have strong personalities tending towards extremes (Egri, 1960). Subtle nuances of a character's personality are captured by the actions that character performs that are inconsistent with its traits. However, such subtle nuances of a character's behavior are not necessarily captured in the domain representation of the initial world state and would instead be captured with complex heuristics (see Section 5.3.5).

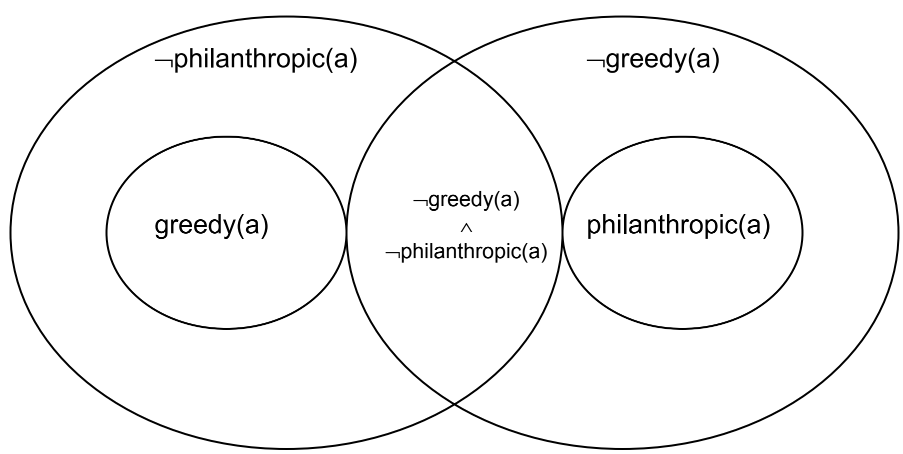

Figure 5.2. The set relationship of polar opposite personality traits.

By relating certain traits as polar opposites, the personality model opens the field up to possibilities where a character does not have to suffer from an extreme set of traits. Polar relationships also establish a means for inferring inconsistencies in character personalities. The framework allows for a trait to be in a polar relationship with more than one other trait. The relationship between greedy and philanthropic traits has already been established. Suppose the personality model contains another trait, misanthropic, which has a polar relationship with the philanthropic trait. If a character is philanthropic, then that character is necessarily not greedy and not misanthropic. But a character can be greedy and misanthropic if that character is not philanthropic.

A personality model, made up of the trait descriptors chosen by the domain engineer, is provided as an initialization parameter to the fabula planner. The personality model definition consists of a set of trait descriptors that can be used to define a character's personality. In addition, polar relationships are declared by pairing trait descriptors. An example of a personality model definition is shown in Table 5.1. The example personality model uses the traits defined for FFM plus three extra traits: philanthropic, greedy, and misanthropic. The traits are given as a list. The polar relationships are provided separately. The example personality model is roughly based on the FFM, except the distinction between conscientiousness versus lack of conscientiousness is a binary relationship instead of a polar relationship. That is, `conscientiousness` is defined as a trait, but `lacking-conscientiousness` is not. Thus a character can have conscientiousness or not have conscientiousness and `¬conscientiousness` is functionally equivalent to lacking conscientiousness. Of course the domain engineer could easily have added a lacking-

Table 5.1. An example of a personality model definition.

```
(define (personality-model)
   :traits (philanthropic greedy misanthropic
            openness closedness conscientiousness
            extraverted introverted
            agreeable disagreeable
            neurotic emotionally-stable)
   :polarities ((openness closedness) (extraverted introverted)
                (agreeable disagreeable) (neurotic emotionally-stable)
                (philanthropic greedy)
                (philanthropic misanthropic)))
```

conscientiousness trait if she wanted to be able to distinguish between `¬conscientiousness` and `lacking-conscientiousness`. The personality model has additional traits, philanthropic, greedy, and misanthropic, that are not part of the FFM to illustrate how a trait descriptor can be in a polar relationship with more than one other trait descriptor at a time. Philanthropic opposes greedy and also opposes misanthropic.

The personality model defined by the human author is not a description of any one individual character in the story world. Instead the personality model describes the vocabulary with which a character can be described. Polarities constrain the vocabulary by specifying descriptors that cannot be simultaneously used to describe a character. Each story world character has a distinct personality which is represented in the initial state of the plan by selecting consistent sets of descriptive terms from the personality model. In describing a character's personality, the human author is making a particular commitment. That is, by describing a character as having a particular personality trait, the human author is asserting her belief that if the audience were to observe the character over a substantial period of time, they would come to believe that that character can be described with that particular trait.

5.3.2. Informing the Planner about Operator Appropriateness

The personality traits of all characters are given as part of the initial state of the world. Assigning personality traits to characters is meaningless unless they have some affect on the way in which operators are selected by the planner. There needs to be some relationship between plan operators and the traits defined by the personality model. Plan operators that are character actions – operations that are regarded to be performed by a story world character – have at least one parameter that corresponds to the character that is the action's actor. One way to correlate character actions to the personality model is to constrain each action in such a way that only characters with prescribed traits can be the actor. For example, the character action `Donate(?char, ?thing, ?charity)` should only be performed by characters that are philanthropic unless a character is acting inconsistently. That is, `(philanthropic ?char)`, should be true in the world where the unbound variable in the action specification unifies with the unbound variable in the trait sentence. Given that personality traits are encoded as part of the initial state of the world, this condition will be satisfied by the initial step of the plan if the character referenced by the unbound variable is in fact philanthropic.

One way to ensure consistency between a character action and the personality of the acting character is to encode statements about the personality traits of the acting character are encoded as preconditions of the action. This would require the planner to add causal links from the trait-defining sentences in the initial state to each character action, equating the required traits for the acting character with plan soundness. The precondition approach, however, causes plans to be discarded if any preconditions cannot be satisfied by the initial state or by any other preceding action whose effects change a character's personality traits. The acting character of any action must have the required traits or the plan will be incomplete. In this approach, no character can perform an action that is inconsistent with its traits.

Encoding character traits into preconditions is an overly strong constraint on the planner. Instead of strongly enforcing which traits a character has in order to perform an action, an alternative approach is to recommend which traits a character should have in order to perform an action and be considered acting consistently. The distinction is that the character does not *necessarily have* to possess certain traits to perform an action, merely that the character *should have* certain traits. If the character does not possess those traits, then the character is considered acting inconsistently, but is still allowed to act. As to whether an inconsistent action detracts from the audience's perception of a character as believable depends on the overall history of actions that character performs throughout the story. The recommendation approach is implemented by the framework for integrating a personality model with the fabula planner. Each character action, in addition to the preconditions that determine whether necessary conditions in the world exist for the action to be executed, has an additional `recommendation` field which suggests propositions about the acting character's traits. Recommendations work similarly to preconditions in that they are satisfied by causal links from prior plan steps that have effects that unify with the recommendation condition. The difference is that recommendation conditions do not need to be satisfied for a plan to be considered sound and complete.

Personality recommendations are encoded directly into action schemata in the fabula planner's action library. Since the planner can choose to not satisfy the personality recommendations on instantiated action schema, it is possible for planner to find a solution plan even if action schemata have inconsistent personality recommendations. If an action schema has inconsistent personality recommendations, such as recommendations for p and $\neg p$ or recommendations for p and q where p and q are polar opposites, the planner needs only to ignore one recommendation or both to succeed in finding a plan. Ignoring a recommendation is tantamount to declaring a character as acting inconsistently, although in this case, it would impossible for a character to act consistently because of an internal contradiction. To ensure that internal contradictions cannot occur, the personality model definition, such as the one shown in Table 5.1, is provided to the fabula planner as an initialization parameter. The schemata in the action library are checked against the personality model definition. Only personality traits can be recommended for an action schema so that the planner is not able to ignore propositions that are required for the soundness of the plan. Action schemata are also checked to make sure they do not have impossible combinations of recommendations, such as p and $\neg p$ or p and q where p and q are polar opposites.

5.3.3. Enforcing Character Consistency

Actions can have recommendations in addition to preconditions. The recommendations are satisfied as if they were preconditions except that recommendations do not need to be satisfied for a plan to be considered sound and complete. Accordingly, the partial order planning algorithm is augmented with a new type of decision point: the *open recommendation flaw*.

> **Definition 5.1 (Open recommendation flaw).** *An open recommendation flaw is a tuple,* $\prec$s, p$\succ$, *where* s *is a plan step and* p *is a recommendation of* s *that has not been previously considered.*

The partial order planning algorithm is split into two parts, causal planning and recommendation planning, corresponding to the type of flaw that is repaired. Causal planning repairs open condition flaws; there is a step in the plan with a precondition that is not satisfied. There are no changes to the way in which open condition flaws are repaired relative to the algorithm specified in Figure 3.2. Recommendation planning repairs open recommendation flaws. The modified partial order planning algorithm is shown in Figure 5.3. Open recommendation flaws are handled in one of two ways. One possibility is that the open recommendation is satisfied by a causal link. In most cases, the causal link will originate from the initial step because this is where character traits are represented. However, it is possible for a causal link to originate from another step in the plan that has an effect that unifies with the open recommendation. Such a plan step affects the character's personality and represents a change in the character's traits.

The other possible way of handling an open recommendation flaw is to ignore the recommendation. That is, the open recommendation is left unsatisfied, the flaw annotation is removed from the plan and no further modifications are made the plan structure to address

POCL-R ($\prec S, B, O, L \succ, F, \Lambda$)

I. **Termination**. If O or B is inconsistent, fail. If F is empty return $\prec S, B, O, L \succ$.

II. **Plan Refinement**. Non-deterministically do one of the following.

- **Causal planning**
  1. **Goal selection**. Select an open condition flaw $f = \prec s_{need}, p \succ$ from F. Let $F' = F - \{f\}$.
  2. **Operator selection**. Let s_{add} be a step that adds an effect e that can be unified with p (to create s_{add}, non-deterministically choose a step s_{old} already in S or instantiate an action schema in Λ). If no such step exists, backtrack. Otherwise, let $S' = S \cup \{s_{add}\}$, $O' = O \cup \{s_{add} < s_{need}\}$, $B' = B \cup$ bindings needed to make s_{add} add e, including the bindings of s_{add} itself, and $L' = L \cup \{\prec s_{add}, e, p, s_{need} \succ\}$. If $s_{add} \neq s_{old}$, add new open condition flaws to F' for every precondition of s_{add}.
  3. **Threat resolution**. Performed in the standard way.
  4. **Recursive invocation**. Call POP-R($\prec S', B', O', L' \succ, F', \Lambda$).
- **Recommendation planning**
  1. **Goal selection**. Select an open recommendation flaw $f = \prec s_{want}, p \succ$ from F. Let $F' = F - \{f\}$.
  2. **Operator selection**. Non-deterministically do one of the following.
     - Let s_{add} be a step that adds an effect e that can be unified with p (to create s_{add}, non-deterministically choose a step s_{old} already in S or instantiate an action schema in Λ). If no such step exists, backtrack. Otherwise, let $S' = S \cup \{s_{add}\}$, $O' = O \cup \{s_{add} < s_{want}\}$, $B' = B \cup$ bindings needed to make s_{add} add e, including the bindings of s_{add} itself, and $L' = L \cup \{\prec s_{add}, e, p, s_{want} \succ\}$. If $s_{add} \neq s_{old}$, add new open condition flaws to F' for every precondition of s_{add}.
     - Ignore the recommendation and do nothing.
  3. **Threat resolution**. Performed as in II.3 in the POCL algorithm (Figure 3.2).
  4. **Recursive invocation**. Call POCL-R($\prec S', B', O', L' \succ, F', \Lambda$).

Figure 5.3. The personality framework algorithm.

the recommendation. The ability to ignore a recommendation preserves the completeness of the algorithm with respect to the repertoire of actions that a character should be able to perform in the world. When a character action is in the plan but has a trait recommendation for the character that is inconsistent with the character's traits in the initial state of the world, the planner has two choices. It can find another action that changes the character's traits to match the recommendation, or it can ignore the recommendation. Ignoring a recommendation, in this case, effectively means that the story world character is acting inconsistently with respect to its declared personality.

5.3.4. Personality Change

People change over time. In the course of a story, a character may have experiences that cause that character to change in such a way as to adopt new tendencies. The framework for a personality model allows for a character's personality traits to change through the instantiation of an action schema with personality trait descriptors as action effects. Such an action has the effect of specifying a new personality trait for a character or revoking a personality trait for a character. Due to the discrete nature of operations in a plan personality change – like any change to the state of the story world – is abrupt. Before the application of an action, a character can be described as having one trait. After the application of that action, that character is described as having a different trait. The nature of personality change in a story plan is a simplification; in real life, ones personality changes gradually as an accumulation of experiences causes a subtle shift in ones tendencies. Gradual change is possible in planning domains, but requires an extremely complicated world state representation and corresponding operator library. If the story plan operators are at a relatively high level of abstraction, then discrete personality change may in fact be appropriate. For example, if one operator describes how a hawkish character goes to war, witnesses the brutality of combat, and is wounded in action, then it may be appropriate for the effects of that single operator to change the character to a pacifist.

5.3.5. Heuristics

Recommendations can be ignored, meaning that they are not satisfied by causal links in the complete plan structure. Since every recommendation of every plan step can be ignored, it is possible that a solution plan found by the fabula planner has no satisfied recommendations. In fact, such a solution plan may be preferred over other plans with satisfied recommendations because the fewer number of links means the fewer number of possible causal threats. In most situations, however, the links that satisfy open recommendations will not cause causal threats unless there are many action schemata that cause personality changes.

Satisfaction of a recommendation implies character consistency and is preferable because it implies that a character is behaving believability. Ignored recommendations can be classified into one of two scenarios. The first scenario for why a recommendation is ignored is because the recommendation cannot be satisfied. The character simply does not have the recommended trait and there is no action that can be performed in the world that will change the character so that he has the trait. This scenario implies that the character is acting inconsistently and although inconsistent character behavior is not preferable, it is possible. The second scenario for why a recommendation is ignored is because the recommendation can be satisfied but is not satisfied. That is, the plan node in which the open recommendation flaw is ignored has sibling plan nodes in which the open recommendation is satisfied. This scenario is problematic for the fabula planner because it means that the character is acting consistently but that the plan structure is not updated to reflect that fact. The scenario cannot be eliminated without affecting completeness of the algorithm, however. If the plan node is not a leaf node, it may be possible that a future decision point causes an action to be inserted into the plan that changes the character's personality to be inconsistent with the ignored recommendation. In this case, the new interpretation is that the character might have been acting consistently except for the fact that the character's traits were changed and now the character is acting inconsistently.

5.3.5.1. A Domain Independent Heuristic for Preserving Character Consistency

For character believability to be present in a story plan, story world characters should behave as consistently as possible. Given many possible solution plans, the fabula planner should prefer those plans in which the most recommendations are satisfied. A domain independent heuristic function penalizes plans with unsatisfied recommendations. If a recommendation can be satisfied, the heuristic always favors the child plan node that satisfies the recommendation instead of the child plan node that ignores the recommendation. Only when satisfying a recommendation causes an un-resolvable causal threat will the fabula planner backtrack and try the child plan that ignored the recommendation. Any causal threat that caused the other sibling to fail will not be present in this branch of the search space because the causal link to the recommendation is absent. If a recommendation cannot be satisfied, then there is no sibling node that is preferred.

5.3.5.2. Domain Dependent Heuristics that Permit Character Inconsistency

One of the drawbacks of relying on a trait-based model of personality is that every character action recommends the traits that the acting character should have. It is not difficult to think of situations where acting "out of character" is not only acceptable but actually desirable. Certainly if the only way for a character to achieve a goal (either an internal character goal if IPOCL is used in conjunction with the personality model framework or a story outcome goal otherwise) is for the character to perform some action that is inconsistent with the character's defined personality traits, the modified planner will find a solution. However, if there is another way for a character to achieve a goal that does not involve acting inconsistently, that solution will be preferred. A more sophisticated domain-dependent heuristic function is required to recognize situations where acting "out of character" is favorable and plans should not be penalized as normal. For example, a non-gluttonous character stuffs himself with food before embarking on a perilous journey across the desert without provisions. The action appears gluttonous at first, but is later justified. Another example is if a character is being coerced or otherwise controlled by a malicious force, it is rational for that character to act out of character since the behaviors that the character performs are not an indicator of his personality in the sense that someone or something else made him do it.

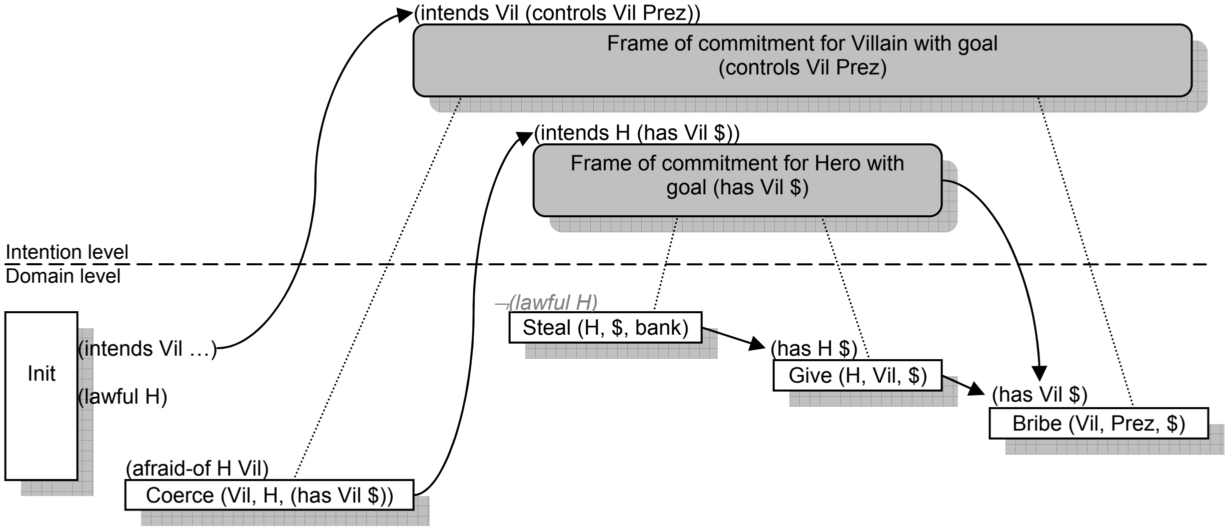

Figure 5.4. A partial plan with an action with an unsatisfied recommendation.

The example introduced in Section 4.5 can be used to illustrate the way in which domain-dependent heuristics can correct oversimplifications in character consistency. A villain threatens the Hero character and uses the threat of force to coerce the Hero to want to achieve the Villain's sub-goal of having money. Although the example is used to illustrate the IPOCL algorithm, the IPOCL algorithm and the personality model framework work together without interference. As Section 4.5 describes, the Hero commits to the goal that the Villain has the money. To achieve this goal, the Hero character steals some money from a bank and gives it to the Villain. Suppose the action schemata in the example used personality recommendations so that only unlawful characters could steal from the bank. Furthermore, suppose that the Hero is lawful. That is, `(lawful Hero)` is declared as part of the initial world state. The scenario so far is shown in Figure 5.4. Conditions in italics are recommendations. The open recommendation on the `Steal` action is that the character doing the stealing should be not lawful. The Hero character is lawful and no action schemata exist that can change a lawful character into an unlawful character. The only way the fabula planner can handle the open recommendation flaw is to ignore it. The domain-independent heuristic described in Section 5.3.5.1 will penalize the plan since it has an unsatisfied recommendation. If there is another way for the Hero to get some money, the planner will pursue that option over the one in which the Hero steals money from a bank. However, from the perspective of the audience, the Hero character's act of stealing does not necessarily appear inconsistent because the Hero is forced against his will into helping the Villain. A domain-dependent heuristic that knows about the `Coerce` action and what it implies can analyze the plan structure and know that it does not need to penalize a plan if the character actions with unsatisfied personality recommendations are motivated by `Coerce`.

A domain-dependent heuristic can determine when a character action that would otherwise be considered inconsistent with that character's personality (e.g. not satisfying a recommendation) is acceptable or even appropriate. Furthermore, a domain-dependent heuristic could also rank the importance of a character's traits. For example, if there are only two complete story plans in the planner's search space, and each plan requires a character to ignore a personality recommendation, the heuristic can determine which inconsistency is less likely to impact the audience's perception of character believability. The ranking between alternative plans may be due to the significance of personality traits to the audience. The ranking may also be due to the degree with which the character actually expresses that personality trait. That is, a character may be described as being greedy, but only mildly so; the binary distinction between greedy and not greedy does not capture such nuance. Another way in which domain-dependent heuristics can capture the nuances of character personality is to suggest certain personality traits as being incompatible. The personality model can describe traits as being polar opposites, but a more subtle relationship exists where it is *unlikely* but possible for a character to exhibit certain traits. For example, greedy and philanthropic do not necessarily have to be polar opposites.

5.4. The Domain-Independent Personality Model as a Theory of Motivation

Recently, personality theory has been adopted as part of motivational psychology (Ford, 1992). Motivational psychologists consider personality as a tendency of an individual to have certain goals. The goals in question tend to be vague, such as the goal to be successful or to be popular. The goal-based personality model of Rizzo et al. (1999) is a computational manifestation of a motivational personality theory. The framework for a domain-independent personality model described in this chapter is a trait-based personality model and does not necessarily conform to any particular field of psychology. Despite the trait-based nature of the framework presented here, a case can be made that the framework supports the motivation-based interpretation of personality. Specifically, the IPOCL algorithm reasons about character goals and the actions that occur in the world that motivates a character to have a goal. The motivating actions whose effects cause characters to have internal goals are fundamentally no different from any other character action. Combining IPOCL with the personality model framework, motivating actions can *recommend* the personality traits that the acting character should have. If the motivating step is one that causes another character that is not performing the action to have a motivation, it can also recommend the personality traits that the influenced character should have. For example, the action `Persuade(?char1, ?char2, ?goal)` – the character bound to variable `?char1` persuades the character bound to variable `?char2` to adopt a particular goal bound to variable `?goal` – can be designed to recommend that the persuading character have the trait of being persuasive and the target character have the trait of being gullible. In this way the goals a story world character commits to are regulated by character personality. Not only are the actions that a character performs consistent with regards to the description of that character's personality, but the intentions that a character forms are consistent with regards to that same description of that character's personality.

5.5. Summary

Character believability relies on consistency of character action. One way to ensure consistency is to provide definitions of story world character personalities and then to evaluate their actions in the story plan against the definition. If the actions that a character performs in the world are consistent with its personality traits, then one can be reasonably certain that that character is acting believably. Because trait theories of personality are based on folk psychological terminology for individual traits, trait theories are both common and plausible. In order to ensure that story world characters are acting consistently, I provide an extension to the POCL algorithm that allows a user-defined personality model to be incorporated into the planning process. Character traits are defined as sentences in the initial state of the world. Actions that are instantiated into the story plan can recommend that the acting character have certain traits. Traits are binary descriptors and characters either have that trait or do not have that trait. Some traits are specified as polar opposites in the personality model, meaning that if a character has one trait, then it does not have any of the polar opposite traits.

The personality trait recommendations operate like open condition flaws and are resolved by causal links that extend from the initial step or from any step in the plan that causes a personality change. However, unlike open conditions, recommendations do not have to be satisfied. The greater then number of recommendations that are satisfied, the more consistently the characters are behaving. Heuristic functions can easily rank plans according to the number of satisfied recommendations. Unfortunately, there are situations where not satisfying a recommendation does not mean the character is acting inconsistently. Situations where one character is forced to act "out of character" can exist. More sophisticated heuristic functions are needed to recognize such situations.

Chapter 6

Initial State Revision

When used within an author-centric approach to story generation, a fabula planner is the sole authority on the story world and the characters in the story world. The only reason that the planner would have any uncertainty as to the true state of the story world would be if the human author specified that there was uncertainty. The *closed world assumption* (CWA) is to presuppose complete and correct knowledge about the world. Since the universe of discourse that describes any world is potentially infinite, the practical implementation of the closed world assumption is to model everything about the world that is known to be true so that everything that is not explicitly stated to be true is presumed false. The closed world assumption allows agents to operate in worlds where there is incomplete information about the state of the world by assuming everything not known is false. Applying the closed world assumption to dramatic authoring yields an interesting twist: a human author has the freedom to invent the story world. If the world given to the planner is a fictional world invented by the human author, the planner can truly assume that everything that is not declared true is, in fact, not true. This distinction is purely semantic and does not impact the way in which planning works. However, it does place a sizable burden on the human author; the human author must completely specify the story world or risk limiting the stories that can be found by the planner.

To illustrate the way in which reinterpreting the closed world assumption can impact planner success, consider the following example. Suppose there is a story world with two principal characters: a secret agent and an international terrorist mastermind. The terrorist lives in a fortress which is heavily guarded such that no person can enter the fortress while armed. The terrorist mastermind is alive in the initial state of the world and the outcome of the story is that the terrorist mastermind is no longer alive. Further suppose the human author specifies the story world such that no lethal weapons are allowed to enter the fortress. Such a story world may very well model the domicile of a paranoid international terrorist mastermind. This description of the story world disallows any story that involves the secret agent entering the fortress and assassinating the terrorist, for the secret agent would be unable to carry a weapon past the guards. Furthermore, suppose that the fabula planner implements the IPOCL planning algorithm and cannot find any plan in which the terrorist is motivated to leave the fortress (indeed, the terrorist is motivated *not* to leave the fortress). In this case, the fabula planner can find no complete story plan for this story world. To circumvent problems caused by the way in which the story world was specified, the human author must reason about the description of the world and its impact on the quality and existence of solution plans. The human author would have to have a mental model of the planning algorithm to realize the problem beforehand or to understand the cause of failure afterwards. I wish to decouple the fabula planner from the human author's presuppositions about solution story plans (for discussion on ways in which the human author can influence the solution story plan, see Section 7.1.1).

One way to resolve the issues that arise due to completeness of story world state is to allow the human author to artificially introduce indeterminism into the story world. Artificially introducing indeterminism means that the human author specifies that some facts about the story world state are not determined ahead of time. The closed world is too limiting for a fabula planner that allows indeterminism because the planner would assume that every fact that is not determined by the human author is false. Planning when the state of the world is indeterminate is called *open world planning*.

6.1. Open World Planning

There are two basic approaches to planning when the world state is indeterminate. The first is contingency planning, in which a planner develops a conditional plan that accounts for every contingency that could arise. The outcome of sensory actions determines which part of the plan to execute. The second approach is conformant planning, in which a planner develops a single, non-conditional plan that does not rely on sensory actions. A conformant plan will succeed no matter which possible world the agent is actually in. Successful conformant plans manipulate the environment to reduce uncertainty (Smith & Weld, 1998). For example, consider the "bomb in the toilet" problem (McDermott, 1987) in which there are two packages, one of which has a bomb in it. Dunking the package with the bomb in the toilet renders the bomb disarmed. If the agent does not know which package has the bomb, then the agent is in one of two possible worlds: one in which the bomb is in the first package and one in which the bomb is in the second package. A conformant plan would handle this situation by generating a plan in which both packages are dunked in the toilet. Although this action does not resolve uncertainty about which possible world the agent is in, the action removes uncertainty the agent has about conditions in the world that are unknown to it. Conformant planning is only possible when there are operations that the agent can perform in the world to resolve uncertainty.

Contingency planning is not an appropriate technique for a fabula planner with incomplete information about the world. The fabula planner is not operating in service of a single agent that is situated in the world and can perform sensing actions. It is not the purpose of the fabula planner to discover the way the world really is.

Conformant planning is more suited for use as a fabula planner because this technique does not require sensory actions to determine the true state of the world. However, plans that resolve uncertainty by manipulating the world via plan actions into a state where the plan does not rely on any indeterminate state information can result in unwieldy story plans. A conformant planner can either avoid making references to facts about the world that are not determined or transform the state of the world through actions into some state where there are no indeterminate propositions. It is one thing for a story world character to dunk both packages in a toilet because he, faced with incomplete knowledge about the world, does not want to be blown up. It is another matter altogether for all packages in the world to be dunked in toilets so that the planner can construct plans without indeterminism regarding packages and explosions. In the first case, uncertainty about the true state of the world belongs to the character whose perceptions of the world are modeled by the domain; the true state of the world is known to the planner but unknown to the characters modeled by the planner. In the second case, there are propositions about the world whose values are unknown to the planner. All indeterminism is resolved by disarming anything that *might* be a bomb; the story world characters were never in any danger. Whether or not a character is in danger is irrelevant since a fabula planner has complete authority in the world and characters are nothing more than effectors of change; a character is not going to be harmed or be killed unless there is some goal or sub-goal that requires the character to be harmed or killed.

A further limitation of conformant planners in the context of fabula planning is that the ability of conformant planners to construct solutions to planning problems containing indeterminate state descriptions is dependent upon the planning problem's action operators. Only those indeterminate propositions that can be altered by operators available to the planner can be forced within a plan into one state or another. Consider the situation where the personality of a character is not fully determined and there is no determination about whether a character is greedy or philanthropic. Assuming there is no operator that can change a character's personality from greedy to philanthropic or vice versa, the only solution a conformant planner can consider is to avoid generating any story in which that character either shows greed or philanthropy.

6.2. Initial State Revision Planning

An alternative to contingency planning and conformant planning for story worlds with indeterminate states is for the planner to search for a possible world in which it can tell the best story. An indeterminate world state defines a set of possible worlds, each world in the set differing on their truth assignments to the indeterminate sentences. While it is not a sound strategy for an agent in an indeterminate world to construct a plan that arbitrarily assumes it is operating within one of those possible worlds, a fabula planner is in the unique position of being able to select which possible world for which it constructs its plan. The fabula planner is the sole authority on the story world and all that occurs within it, in the same sense that a human author is the sole authority of the story world she creates. From the perspective of the audience, the story can take place in any one of a universe of possible worlds. The world as understood by the audience is not necessarily the same world understood by the author since the audience only learns about the story world through descriptions and through observable actions of characters situated in the world (Ryan, 1991). Given the audience's uncertainty about the story world (which differs from the indeterminate world state representation of the planner, if at all), the audience can consider the actual story world to be one of a set of possible worlds until a fact is that they were once ignorant of. Over time the set of possible worlds that the audience believes the story is set in converges on the possible world that the author intends, assuming the narration is reliable (Ryan, 1991). In contrast to human authors, in the work described here, a fabula planner cannot inaccurately represent the story world.

A fabula planner given an indeterminate initial world state can non-deterministically choose any possible world for the story to be set in as long as the information about the story world that the audience learns during the telling of the story does not contradict the planner's choice of possible world. For example, suppose the definition of a story world character is indeterminate with respect to whether that character is greedy or philanthropic. If the best story that the fabula planner can come up with involves that character acting philanthropically, it is sufficient for a fabula planner to declare to itself that that story is set in a world in which the character is philanthropic and has always been philanthropic as long as the audience did not already know that the character is not philanthropic. There need not be an operation in the world that makes the character philanthropic (as in conformant planning). Instead, the philanthropic nature of the character is determined, reactively, to have been part of the initial state of the story world. The audience learns about the character's trait when the character performs his first philanthropic act. The fabula planner must then take special care not to subsequently change this declaration about the initial state of the world.

How can the notion of artificial introduction of indeterminism into the world and the selection by a planner of the possible world in which the story is told help our human author in the secret agent/terrorist mastermind example? The problem is that, without modeling the process of plan search, the world in which there are no weapons inside the fortress seems reasonable to the human author who sets up the initial world state. This initial world state declaration, however, renders it impossible for the fabula planner to find a plan in which the outcome – the terrorist is not alive – becomes true. Suppose that there are aspects of the story world that the human author wants to declare in a specific way and other aspects that the human author has no definite opinion about. For example, the human author may insist that the secret agent initially be outside of the fortress and the terrorist mastermind be inside the fortress, but the human author has no specific preference about the initial location in the story world of a particular gun, `gun1`. The human author declares the location of `gun1` to be indeterminate. The fabula planner assumes responsibility for resolving any indeterminism about the weapon's location if that weapon is referenced in the fabula plan.

Initial State Revision (ISR) planning is a technique whereby some finite amount of indeterminism in the initial world state is resolved by searching, in a least-commitment manner, for a possible world in which the story is set. That is, the planner assumes it is in all possible worlds consistent with its representation of the world until it needs to commit to a particular value of an undetermined sentence describing the world. Indeterminism is introduced by the human author who provides an incomplete description of the initial state of the world. ISR modifies the closed world assumption (CWA) to be the assumption that the true state of the world is completely known *except for sentences that are explicitly declared to be undetermined.* The modified CWA separates sentences whose truth values are undetermined into two classes: those that are assumed, for better lack of knowledge, to be false and those that are not assumed to be false. ISR, consequently, classifies all knowledge about the world into three sets: known true ($\mathcal{T}$), false ($\mathcal{F}$), and undetermined ($\mathcal{U}$). $\mathcal{T}$ is the set of atomic ground sentences that are known to be true. $\mathcal{F}$ is the set of atomic ground sentences that are known to be false or are assumed to be false because their true value is not specified by the human author. The semantic interpretation of sentences in $\mathcal{F}$ is the set of sentences whose negations are true. The $\mathcal{T}$ and $\mathcal{F}$ sets are present under the conventional CWA treatment. $\mathcal{U}$ is the set of atomic (but not necessarily ground) sentences whose truth values are undetermined but are not assumed to be false. For a fabula planner, $\mathcal{U}$ is the set of knowledge about the state of the world that the human author has explicitly designated having no preference over. The fabula planner assumes that $\mathcal{T}$ and $\mathcal{F}$ are known while the possible combinations of values of the sentences in $\mathcal{U}$ describe the ways in which possible worlds can differ.

Definition 6.1 (ISR Initial State). *The initial state of the world is a tuple,* $\prec\mathcal{T}, \mathcal{U}\succ$, *where* $\mathcal{T}$ *is the set of atomic ground sentences that are known to be true and* $\mathcal{U}$ *– called the undetermined set – is the set of atomic sentences that are undetermined and not assumed false. Atomic sentences in the universe of discourse that are not in* $\mathcal{T} \cup \mathcal{U}$ *are assumed false.*

6.2.1. Satisfying Open Conditions with Sentences in the Undetermined Set

ISR planning is an extension of the conventional POCL algorithm. An open condition on plan step, s_{need}, is resolved by finding some step, s_{add}, which has an effect that unifies with the open condition. Step s_{add} can be an existing step in the plan – in which case the initial step, s_0, is a possibility – or a step newly instantiated from an action schema in the action library. The initial step, s_0, is a step with no preconditions and the initial world state as effects (Penberthy & Weld, 1992). Step s_0 is necessarily ordered before all other plan steps. In ISR planning, the effects of the initial step are broken into sets $\mathcal{T}$, $\mathcal{F}$, and $\mathcal{U}$ although in implementation, $\mathcal{F}$ is implied as the universe of discourse minus the union of $\mathcal{T}$ and $\mathcal{U}$. Sentences in the initial state are atomic. Otherwise, sentences can be non-atomic although non-atomic sentences can only use the negation operator. Negated conditions unify with atomic sentences in $\mathcal{F}$ since the semantic interpretation of those sentences is that they are negated. An open condition can unify with sentences in any set, including the undetermined set in the initial state. When an open condition p unifies with a sentence q in the undetermined set, a causal link between s_0 and s_{need} is established as normal and, additionally, sentence q becomes determined. If p is not negated, q is determined to be true and moved into set $\mathcal{T}$, otherwise q is determined to be false and moved into set $\mathcal{F}$. By using q to satisfy an open condition, the planner has in effect committed to a set of possible worlds in which the value of q is determined.

As in conventional POCL planning ISR flaw repair is a process of plan revision. A plan node in the search space in which the open condition on s_{need} is satisfied by sentence q that was once in the undetermined set is distinguished from any other sibling plan nodes in that it reduces the set of possible worlds in which the story might be taking place. For example, if

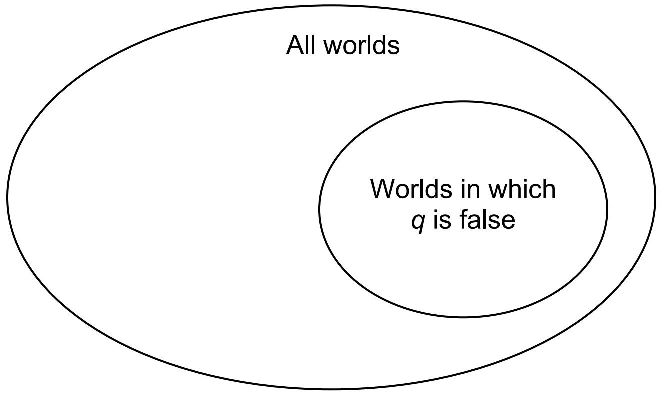

Figure 6.1. Set relationship of possible worlds given propositions with indeterminate truth values.

the open condition is established by determining that $\neg q$ is true in the initial state, the child plan node exists in a world in the set of possible worlds in which q is false before the first plan step occurs. No other sibling node can satisfy the open condition with a causal link from the initial step because $\mathcal{T}$, $\mathcal{F}$, and $\mathcal{U}$ are disjoint sets of sentences; sentence q remains in the undetermined set in the sibling plans. Less indeterminism exists in one sub-tree in the search space because in this one sub-tree all possible worlds in which q is true have been eliminated as possibilities, as shown in Figure 6.1. It is not necessary to reduce the set of possible worlds to one single possible world in which the values of all sentences are known; only the sentences that come into effect in the plan steps need to have known values.

The situation where indeterminism is reduced by committing to a set of possible worlds where an undetermined sentence in the initial state becomes determined is called an *initial state revision* because it alters the description of the initial state of a plan and its successor nodes in the search space. In effect, the planner constrains the set of possible worlds that a story might be told in by committing to a value of a sentence that was previously undetermined. Children of the plan node in which the initial state is revised are all members of the set of possible worlds for which that initial state holds. Figure 6.2 shows a plan space beginning with the empty plan node, p_0. The initial state of p_0 specifies that the value of sentence x is undetermined. Plan node p_2 has an open condition in which $\neg x$ needs to be established for step a. There are three ways to repair the open condition: instantiate step b, instantiate step c, or reuse the start step. Reusing the start step, as in plan p_6, requires that the initial state be revised. Since the value of x is undetermined in the parent node, the value of x

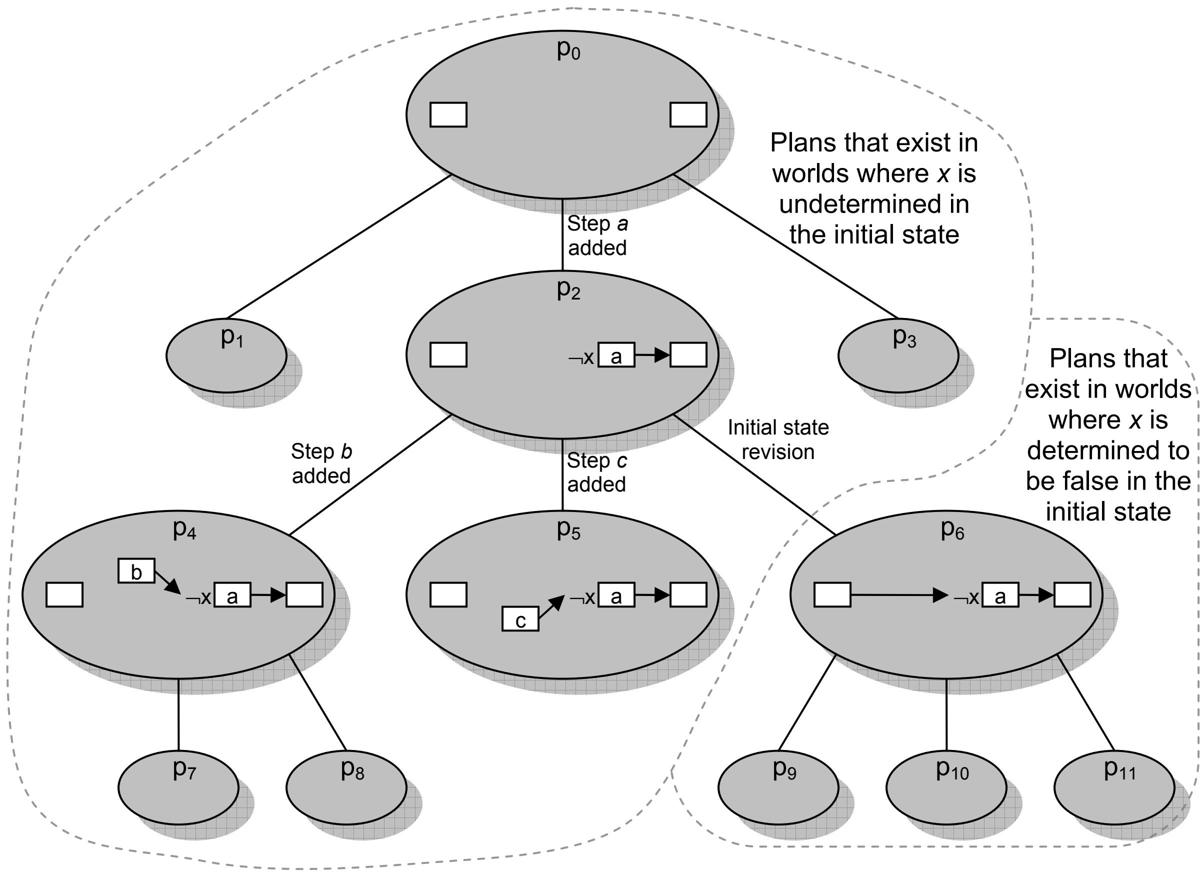

Figure 6.2. Relating possible worlds to plans in the plan search space.

is determined to be false in the initial state and a causal link is extended from the initial step to step *a*. Plan node p_6 is a plan in which *x* is determined to be false, while its siblings, p_4 and p_5, are plan nodes in which the value of *x* remains undetermined. The children of p_6 inherit the initial state of p_6 in which *x* is determined to be false.

6.2.2. Simplifying the Representation of the Undetermined Set

Any indeterminism in the story world is explicitly introduced by the human author. Introducing indeterminism into the story world can relieve the human author from some responsibility for describing a story world in which a story that proves a premise can be found. Introducing indeterminism, however, does not necessarily reduce the amount of effort required for the human author to describe a story world. In some situations, introducing indeterminism can actually increase the number of sentences that are needed to represent the story world (or more accurately, a set of possible story worlds). For example, suppose that there is a particular object that the human author wants to be in the world and there are *n* discrete locations in the story world that the object can be at. Furthermore, suppose that the human author has no strong opinion about which of those *n* locations the object should be at in the initial state of the world and wants to leave the object's location indeterminate. Since the world is described using sentences that have binary truth values, the human author would need to explicitly define *n* sentences in the undetermined set: `(at object place`$_1$`)` … `(at object place`$_n$`)`. Clearly the difficulty of describing indeterminism in the story world increases as the complexity of the story world increases.

To simplify the task of introducing explicit indeterminism into a story world, the fabula planner supports ungrounded sentences in the undetermined set. An ungrounded sentence has one or more variable terms. For example, `(at object ?place)` where `?place` is a variable is an sentence signifying that `object` is at some unspecified location. ISR allows both ground and ungrounded sentences in $\mathcal{U}$. Having an ungrounded sentence in $\mathcal{U}$ is equivalent to having a ground sentence in $\mathcal{U}$ for every unique binding of variables to ground terms. Allowing ungrounded sentences makes it easier for the human author to specify indeterminism in initial state of the story world because the human author does not need to list a sentence in $\mathcal{U}$ for every possible undetermined sentence.

When an open precondition of an action unifies with an ungrounded sentence in the undetermined set, a set of ground sentences – one for every unique binding of variables to ground terms – is substituted for the ungrounded sentence. Which ground terms are valid variable bindings is determined by constraints on the parameters of the action schema. The ground sentence that unifies with the open condition is determined to be true while the remaining ground sentences remain undetermined.

6.2.3. Mutual Exclusion

It is often the case that propositions are mutually exclusive, meaning that if one is true then the other cannot be true. Mutual exclusion applies to negations; sentences p and $\neg p$ cannot be true at the same time. Mutual exclusion also applies to other sentences that are not negations of each other. For example, an object cannot be in more than one place at a time.

So if `(at object place`$_1$`)` is true, then `(at object place`$_2$`)` and `(at object place`$_3$`)` and so on cannot be true. Mutual exclusivity becomes an issue in ISR planning when the undetermined set $\mathcal{U}$ contains atomic sentences that should never be simultaneously true in the initial state of the world. If the planner does not know that two sentences are mutually exclusive, the possibility exists for the planner to determine both sentences to be true in the initial state. To prevent this situation from arising, ISR utilizes *mutual exclusion sets* (or *mutex sets*).

> **Definition 6.2 (Mutual exclusion set).** *A mutual exclusion (mutex) set is a set of sentences such that no two sentences can both be true in the initial state.*

The ISR planner maintains a collection of mutex sets. If an initial state revision occurs in which a previously undetermined sentence is determined to be true then that sentence whose value is newly determined. That is, the previously undetermined sentence is moved from $\mathcal{U}$ to $\mathcal{T}$ if the condition being satisfied is not negated or is moved from $\mathcal{U}$ to $\mathcal{F}$, otherwise. The sentence (negated if it was determined to not be true) is checked against all mutex sets. If the sentence is a member of a mutex set, then all other sentences in the mutex set are determined to be not true. That is, for those sentences that are not negated in the mutex set, the corresponding atomic sentences in $\mathcal{U}$ are moved from to $\mathcal{F}$ and for those sentences that are negated in the mutex set, the corresponding atomic sentences in $\mathcal{U}$ are moved to $\mathcal{T}$. The mutual exclusion rule only comes into effect when a sentence whose value was previously undetermined is newly determined to be true. When a sentence whose value was previously undetermined is newly determined to be false, the values of other sentences that are mutually exclusive remain undetermined because more than one sentence in a mutex set can be false.

The sentences in a mutex set can be negated. For example, the mutex set {¬`(open door`$_1$`)`, ¬`(open door`$_2$`)`} implies that, if the state of the two doors are undetermined, then the story world in which both doors are initially closed is not a possibility. Mutex sets only apply to the description of the initial world state; actions are allowed to change the world in such a way that violates mutual exclusivity relationships. Mutex sets are only for ensuring that the declaration of the initial state of the world is constrained. Presumably action schemata in the action library are defined so that they do not cause illegal world states.

The definition of a plan in ISR is given below.

> **Definition 6.3 (ISR Plan).** *A plan is a tuple,* ≺S, B, O, L, M≻*, where* S *is a set of plan steps,* B *is a set of binding constraints on the free variables in the steps in* S*,* O *is the set of ordering constraints on the steps in* S*,* L *is a set of causal links between steps in* S*, and* M *is a set of mutual exclusion sets.*

The human author is responsible for specifying mutex sets during initialization of the planner. However, the inclusion of ungrounded sentences in the undetermined set makes mutual exclusion more complicated. If the human author specifies an ungrounded sentence in the undetermined set and an open condition unifies with it, the ungrounded sentence is replaced by a set of atomic ground sentences that cover all unique ways of binding the free variables in the sentence to ground terms. One ground sentence becomes true and the others remain undetermined. The human user may want to specify that only one unique binding of variables in an ungrounded sentence can be true in the initial state of the world. For example, if the human author wants to leave the location of an object undetermined, she specifies `(at object ?place)` in the undetermined set. If she wants to ensure that the object cannot simultaneously be in more than one location in the initial state, she needs to specify a mutual exclusivity relationship. One way is to specify a mutex set that exhaustively enumerates the possible ways of binding the free variables in the ungrounded sentences, e.g. {`(at object` $place_1$`)`, `(at object` $place_2$`)`, …}. A simplified syntax allows the human author to specify a mutex set with a single ungrounded sentence, e.g. {`(at object ?place)`}. As with ungrounded sentence in the undetermined set, a mutex set with an ungrounded sentence is equivalent to having a mutex set with a ground sentence for every unique binding of free variables to ground terms. When a sentence that is newly determined to be true unifies with the ungrounded sentence in a mutex set, a set of ground sentence in which each element represents an unique binding of free variables is substituted in the mutex set for the ungrounded sentence; the resulting set is used to determine which sentences in the undetermined set will be determined to be false.

To implement mutual exclusion, the planner is initialized with a planning problem containing the initial state of the world, the goal state, and a set of mutex sets.

Definition 6.4 (ISR Planning Problem). *A planning problem is a tuple,* $\prec$I, G, M, A, $\Lambda\succ$, *where* I *is the initial state of the world according to Definition 6.1.,* G *is the set of atomic ground sentences that partially describe the state of the world after execution is complete,* M *is a set of mutex sets,* A *is the set of agents in the world, and* Λ *is the set of action schemata that describe the actions that agents in* A *can perform in the world.*

From the parameters in the planning problem, the plan search space is initialized with an empty plan node, containing only the initial step, s_0, the goal step, s_∞, and the ordering constraint $\{s_0 < s_\infty\}$. The effects of the start step are separated into two sets, $\mathcal{T}(s_0)$ specifying all atomic ground sentences determined to be true and $\mathcal{U}(s_0)$ specifying all atomic sentences whose truth value is undetermined. A third, implied set is the universe of discourse minus $(\mathcal{T}(s_0) \cup \mathcal{U}(s_0))$ specifying all atomic ground sentences that are determined to be false. The ISR planning algorithm is shown in Figure 6.3.

ISR ($\prec S, B, O, L, M\succ, F, \Lambda$)

I. **Termination**. If O or B is inconsistent, fail. If F is empty return $\prec S, B, O, L, M\succ$.

II. **Plan Refinement**.

1. **Goal selection**. Select an open condition flaw $f = \prec s_{need}, p\succ$ from F. Let $F' = F - \{f\}$.
2. **Operator selection**.
   a. Let s_{add} be one of the following.
      - An instantiation of an action schema in Λ with an effect e that can be unified with p.
      - A step in $S - \{s_0\}$ that adds an effect e that can be unified with p.
      - The start step, s_0, such that $\exists e \mid \{(e \in \mathcal{T}(s_0) \wedge e$ unifies with $p) \vee$ $(e \notin \mathcal{T}(s_0) \wedge e \notin \mathcal{U}(s_0) \wedge e$ unifies with $\neg p) \vee (e \in \mathcal{U}(s_0) \wedge e$ unifies with p or $\neg p)\}$.
   b. Let $S' = S \cup \{s_{add}\}$, $O' = O \cup \{s_{add} < s_{need}\}$, $B' = B \cup$ bindings needed to make s_{add} add e, including the bindings of s_{add} itself, and $L' = L \cup \{\prec s_{add}, e, p, s_{need}\succ\}$. If $s_{add} \neq s_{old}$, add new open condition flaws to F' for every precondition of s_{add}.
   c. If $s_{add} = s_0$ and $e \in \mathcal{U}(s_0)$ then do the following.
      i. Let $s_0' = s_0$. $\mathcal{U}(s_0') = \mathcal{U}(s_0') - \{e\}$.
      ii. If p is not negated then $\mathcal{T}(s_0') = \mathcal{T}(s_0') \cup \{e\}$.
      iii. For all $m \in M$ such that $p \in m$, $\mathcal{U}(s_0') = \mathcal{U}(s_0') - \{q\}$ for all $q \in m$. If q is negated, then $\mathcal{T}(s_0') = \mathcal{T}(s_0') \cup \{\neg q\}$.
      iv. $S' = (S' - \{s_0\}) \cup \{s_0'\}$.
3. **Threat resolution**. Performed as in II.3 in the POCL algorithm (Figure 3.2).
4. **Recursive invocation**. Call ISR($\prec S', B', O', L', M\succ, F', \Lambda$).

Figure 6.3. The ISR algorithm.

6.3. An Example

To illustrate the ISR planning algorithm, consider the story world with the secret agent and the international terrorist mastermind. A diagram of the world is shown in Figure 6.4. The world is distinctly split into two parts: inside the fortress and outside the fortress. The secret agent starts at `headquarters` outside of the fortress. The terrorist is in the `office` inside the fortress. There are documents waiting at a place called `drop-box` that will enable the secret agent to pass through the guards into the fortress. However, the secret agent is prohibited from holding a weapon while passing through the guards. The human author has specified that a weapon, `gun1`, is known to exist but the location of the gun is specified as uncertain. That is, the sentence, `(at gun1 ?place)` is specified in the undetermined set of the initial world state. Additionally, a single mutex set, {`(at gun1 ?place)`} is specified so that the gun cannot be declared to be in more than one location at a time in the initial state of the world.

The outcome of the story is that the terrorist mastermind is not alive. The following is a trace of a single path through the plan search space. The planner non-deterministically satisfies the goal condition ¬`(alive mastermind)` by instantiating a new action `Shoot(secret-agent, mastermind, gun1, office)` in which the secret agent shoots the mastermind with `gun1` in the office of the fortress. There are three preconditions of the `Shoot` action that must be satisfied:

1. `(at mastermind office)` – the mastermind must be at the office,
2. `(at secret-agent office)` – the secret agent must be at the office, and
3. `(has secret-agent gun1)` – the secret agent must have `gun1`.

The open conditions are resolved as ordered. The first open condition can be non-deterministically satisfied by a causal link originating with the initial step. The second open condition is repaired by instantiating a new step in which the secret agent moves from some location in the world to the `office` location. There are six children nodes in the search space that resolve the open condition, one for every location that the secret-agent can be in before moving to the office. Only two branches are valid and will lead to solutions: one in which

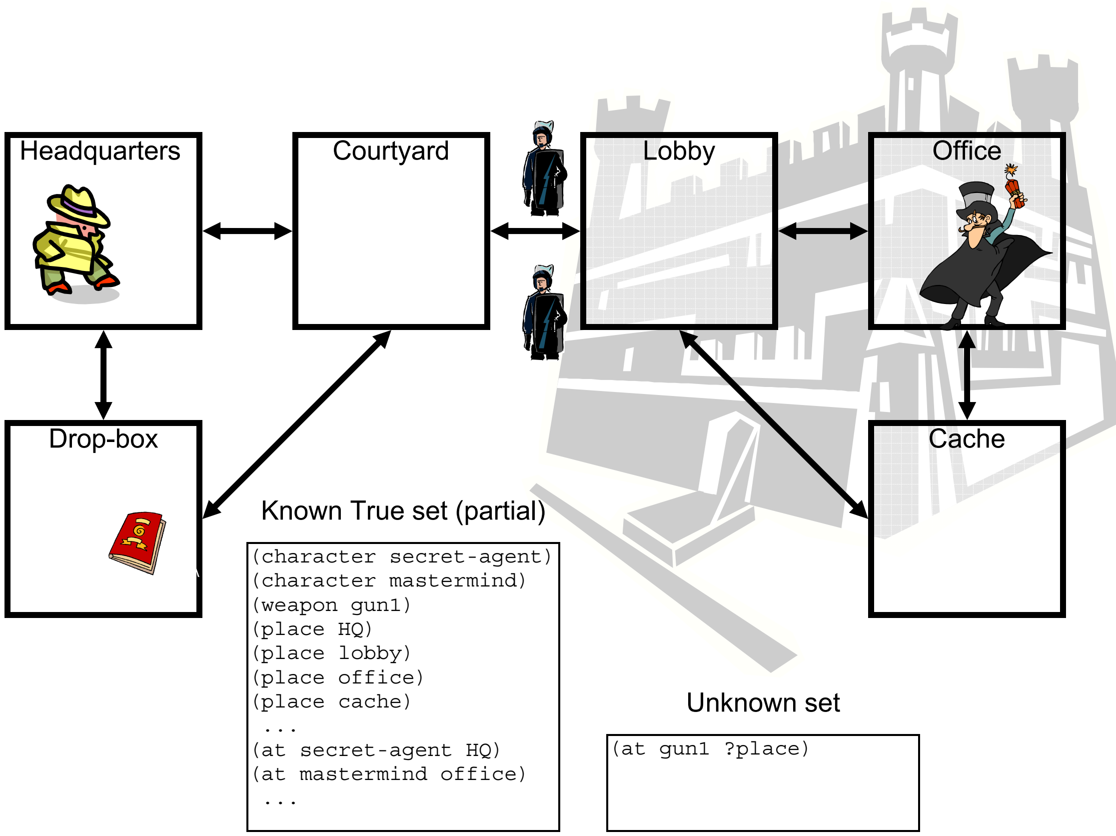

Figure 6.4. Diagram of the secret agent world.

`Go(secret-agent, lobby, office)` is instantiated and one in which `Go(secret-agent, cache, office)` is instantiated. The planner non-deterministically chooses the latter, although both branches ultimately lead to complete plans.

The third open condition is considered next. It is resolved by instantiating a new step in which the secret agent picks up `gun1`. There are six children nodes that repair the open condition, one branch for every location that the secret agent can pick up the gun. The six children nodes repair the flaw by instantiating different variations of the same action schema:

1. `Pick-up(secret-agent, gun1, headquarters)`
2. `Pick-up(secret-agent, gun1, drop-box)`
3. `Pick-up(secret-agent, gun1, courtyard)`
4. `Pick-up(secret-agent, gun1, lobby)`
5. `Pick-up(secret-agent, gun1, cache)`

6. `Pick-up(secret-agent, gun1, office)`

Since the actual location of `gun1` is indeterminate, any of these branches appears to be a possibility because the location of the gun will be determined to be at that location through an initial state revision. However, any of the first three branches will lead to failed plans because if the secret agent picks up the gun outside the fortress, he will be unable to enter the fortress. The remaining three lead to complete plans. The planner non-deterministically chooses the fifth possibility in which the secret agents picks up the gun in the cache.

The action, `Pick-up(secret-agent, gun1, cache)` yields an open condition, `(at gun1 cache)`, indicating that `gun1` must be in the cache for it to be picked up there. There is only one action schema that can satisfy this open condition: `Drop`, in which some character drops the object, implying that that character picks up the object at some prior point in the plan. There will be several children nodes that satisfy the open condition with an instantiation of the `Drop` action schema. There is, however, another way to repair the open condition flaw: an initial state revision determines the location of `gun1` to be at the `cache` and a causal link is extended from the initial step to the `Pick-up` action. The sentence `(at gun1 ?place)` is in the undetermined set, specifying that the location of `gun1` is undetermined. Since the sentence is ungrounded, `gun1` can be in any of the six locations in the world. For the sentence to unify with `(at gun1 cache)`, the free variable `?place` in the ungrounded sentence in the undetermined set is bound to the value, `cache`. Initial state revision determines that `(at gun1 cache)` is a sentence that is true in the initial state of the world. The initial state of the world is constrained by mutual exclusivity pertaining to the initial location of the gun. The ungrounded sentence is removed from the undetermined set so that the planner cannot re-determine the position of `gun1` at any later stage in any sub-tree of the plan node in which the initial state revision occurs. Ground sentences placing `gun1` in the five remaining locations are determined to be not true.

Plan nodes that are successors to the node in which initial state revision occurred as root exist in the possible world in which `gun1` is initially location in the `cache`. Sibling nodes of the plan node in which initial state revision occurred – those that resolve the open condition with an instantiation of the `Drop` action schema – exist in all sets of possible worlds. Continuing

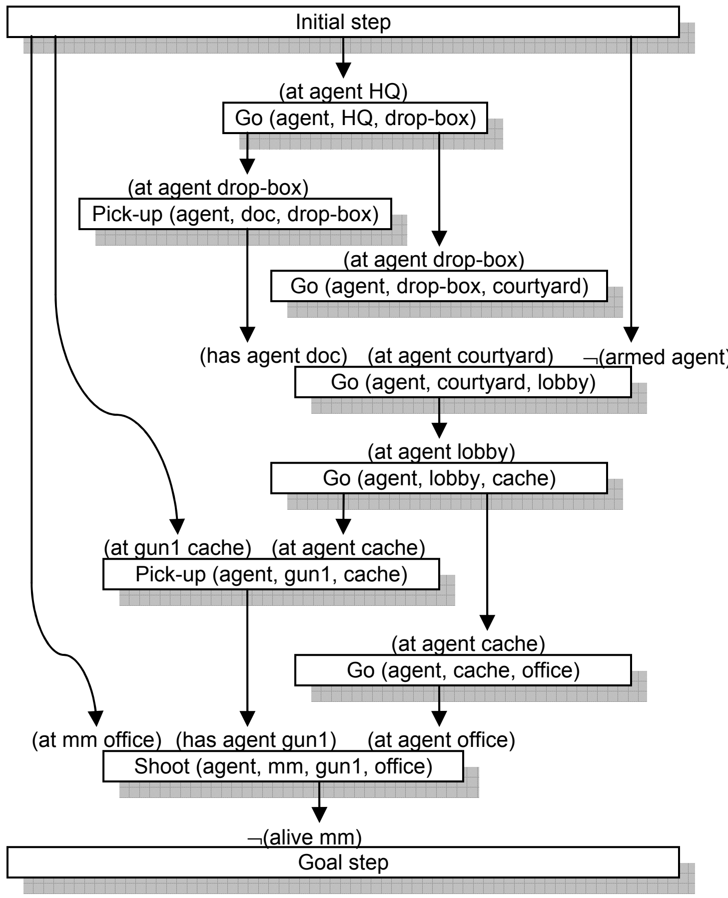

Figure 6.5. A solution plan for the secret agent world.

the example in which the gun is picked up in the cache, the planner has no difficulty finding a plan in which the secret agent retrieves the documents, passes through the guards into the fortress, goes to the cache, picks up the gun, and uses the gun to assassinate the terrorist mastermind. The complete plan for this possible world is shown in Figure 6.5.

6.4. Soundness of ISR plans

Penberthy and Weld (1992) give a proof of soundness of POCL plans based on Pednault's causality theorem (Pednault, 1986). Pednault represented action schemata as sets of *causation* and *preservation* preconditions, collectively called "secondary preconditions." A causation precondition of an action a for some relation R, $\sum_R^a$, specifies all conditions under which a will cause R to be added to the world state. The preservation preconditions of an action a and relation R, $\prod_R^a$, completely specify the conditions under with R is *not* deleted from the world state. Preservation preconditions preserve the truth of *R*. Pednault's causality theorem (Pednault, 1986) is stated below.

Theorem 6.1 (Pednault's causality theorem). *A condition* φ *will be true at point* t *during the execution of a plan if and only if one of the following holds:*

1. *An action* a *is executed prior to point* t *such that*
   a. $\sum_{\varphi}^{a}$ *is true immediately before executing* a.
   b. $\prod_{\varphi}^{b}$ *is true immediately before the execution of each action* b *between* a *and* t.
2. φ *is true in the initial state and* $\prod_{\varphi}^{b}$ *is true immediately before the execution of each action* b *prior to point* t.

Penberthy and Weld use a loop invariant technique for recursive algorithms to prove the soundness of POCL plans. The loop invariant is given below.

Definition 6.5 (The UCPOP loop invariant). *If the sub-goals in the goal agenda* G *are satisfied by plan* P*, then* P *will be a solution to the planning problem* α.

Proof. By corollary 3.29 of Pednault's thesis (1986), one can replace goals in the goal agenda *G* with the causation precondition of steps in plan *P* whose effects achieve those goals and the preservation preconditions of steps that might threaten the goals. By Pednault's causality theorem, if these preconditions are satisfied, the original goals *G* are satisfied. The condition φ and the point t of the causality theorem correspond to the precondition *p* and step s_{need} of the open condition flaw selected to repair by the POCL algorithm (step II.1 of the algorithm in Figure 3.2). The action *a* in the causality theorem corresponds to the new or existing step s_{add}. Instead of requiring that the entire causation preconditions $\sum_{\varphi}^{a}$ be generated and posted as a sub-goal, it is sufficient to post another condition ρ that subsumes the more complex causation precondition formula. That is, $\rho \vdash$

$\sum_{\varphi}^{a}$. ρ is the formula derived from substituting the variable bindings in P to the precondition formulae in a (Penberthy & Weld, 1992).

Penberthy and Weld (1992) demonstrate that the POCL algorithm correctly handles case 1.a of Pednault's causality theorem by recording the satisfaction of condition φ with a causal link from step s_{add} to step s_{need} and correctly handles case 1.b of Pednault's causality theorem by resolving causal threats through promotion, demotion, and separation. Case 2 of Pednault's causality theorem is handled by selecting the initial step s_0 to be the value of s_{add} because if the initial step can be selected to satisfy an open precondition, then the condition is true in the initial state and causal threat prevention ensures that the condition is not deleted before step s_{need} is executed.

The ISR planning algorithm deviates from the POCL algorithm in that certain conditions – those in the undetermined set, $\mathcal{U}(s_0)$ – are indeterminate. If $s_{add} = s_0$ and p unifies with condition e or $\neg p$ unifies with e, then e or $\neg e$ respectively is declared to be true in the revised plan P' and condition e is removed from $\mathcal{U}(s_0)$ in P'. If the loop invariant holds before the iteration of the ISR algorithm in which initial state revision occurs, then it will hold afterwards. The condition $\varphi = p$ is true in the initial state of the revised plan, P' – this is the function of initial state revision, implemented in line II.2.c of the algorithm in Figure 6.3. Furthermore, for every action b prior to s_{need}, $\prod_{\varphi}^{b}$ holds because the ISR algorithm either (a) introduces additional ordering constraints such that b is not ordered prior to s_{need} (demotion) or (b) posts goal $\neg q$ such that $\neg q \vdash \prod_{\varphi}^{b}$ for all such actions b (separation). Causal threat detection and prevention is unaffected by indeterminism because φ is determined in the initial state of P' and the revised initial state of P' is inherited by subsequent invocations of the recursive ISR algorithm. □

6.5. ISR and the Framework for a Domain-Independent Personality Model

ISR provides flexibility to the fabula planner in the case where the human author does not wish to specify an exact story world. Indeterminism in the initial world specification defines

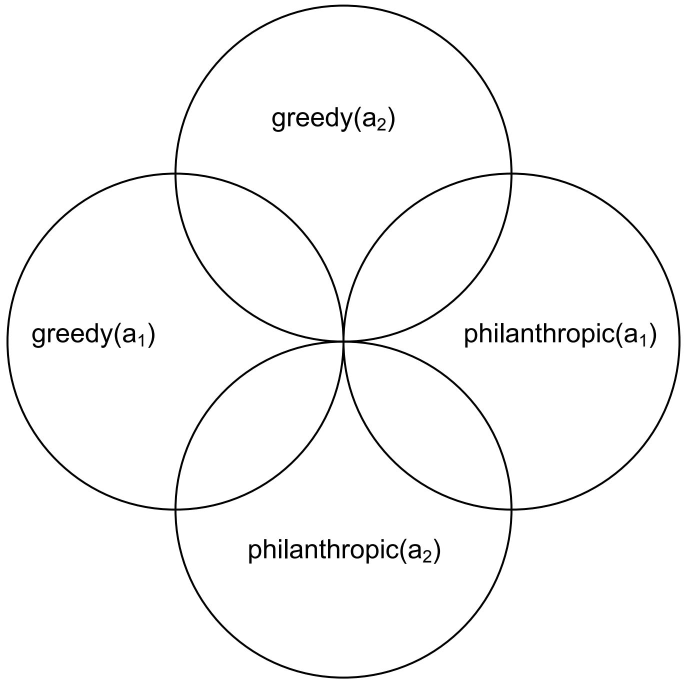

Figure 6.6. Possible worlds in which character definitions are undetermined.

a set of possible worlds that the story is set in. ISR non-deterministically refines the set of possible worlds in a least-commitment manner until a solution plan is found. One reason that the human author might not want to specify an exact story world is to allow for flexibility in the way that characters are determined. The computational model of dramatic authoring, described in Section 3.1.2 models the process of story generation as the search for a set of characters and a plot for which the premise of a story is proven (Egri, 1960). That is, given indeterminate facts about the story world characters, a story planner should not only create a plot consisting of sequence of character actions, but also define the characters in a way that the plot is believable and the outcome inevitable. A least-commitment planner can make commitments to the traits of otherwise indeterminate characters as needed to justify character actions. The process of gradually determining character traits is modeled as the refinement of the set of possible worlds that the story is set in where each possible world differs in how the story world characters are defined. For example, consider a story with just two characters in which the characters *must* either be greedy or philanthropic; the possible worlds that can exist for this story's initial state are shown in Figure 6.6. The number of possible worlds and the complexity of the Venn diagram increases dramatically when one accounts for the possibility of a character being both not greedy and not philanthropic. Since a fabula plan can exist without defining the exact possible world that the story is set in (e.g. some sentences can remain indeterminate), the planner must also account for the possibility that it is determined that a character is determined to be not greedy but not determined whether that character is philanthropic. In this situation, the story is in one of two possible worlds, one in which the character is not greedy and philanthropic and one in which the character is not greedy and not philanthropic. Presumably in such a circumstance, it is not important whether the character is philanthropic or not, as long as the audience knows that the character is not greedy.

The fabula planner meets the requirements of the computational model of dramatic authoring when character personality traits are left undetermined and ISR planning is used to non-deterministically find a possible world in which characters have the traits necessary for a story to prove the premise. In addition to the conventional specification of character actions by lists of preconditions and effects, action operators in our model include personality recommendations; these recommendations advise the fabula planner as to what kind of character is likely to perform the action. Personality recommendations do not need to be satisfied; plan construction can ignore recommendations and build plans in which characters act out of character. However, plans that have a high degree of satisfied recommendations can be characterized as having characters that act most consistently according to their personality definitions. The personality model framework requires character personalities to be defined as binary sentences in the initial state of the world. ISR planning enables the human author to leave the personality trait sentences of characters undetermined. When the planner non-deterministically chooses to satisfy a personality recommendation of a character action in the plan and the recommended condition also unifies with a personality trait in the undetermined set, an initial state revision occurs and the planner commits to a world in which the character performing the action has the recommended trait. Otherwise, the character's personality trait remains indeterminate.

6.5.1. Avoiding Contradictory Possible Worlds

The personality model that is supported by the fabula planner, as described in Chapter 5, consists of binary personality descriptors; a character can either have a personality trait or not have that personality trait. Furthermore, certain personality traits are organized into polar opposites such that if a character has a personality trait, that character cannot also have a polar opposite trait. For example, if the personality traits greedy and philanthropic are polar opposites, then a greedy character must be, by definition, not philanthropic. The character, however, can be both not greedy and not philanthropic. Mutual exclusion, as implemented by ISR, is used to enforce the polar nature of the personality model. When ISR is used in conjunction with the personality model framework, the planner constructs mutex sets directly from the personality model definition for every pair of polar traits. That is, if `(greedy agent)` and `(philanthropic agent)` are sentences in the undetermined set, then the planner automatically creates a mutex set containing those two sentences, e.g. {`(greedy agent)`, `(philanthropic agent)`}.

It is possible for the human author to define some traits of a character to be indeterminate and other traits of that same character to be determined. In this case, the use of mutex sets cannot prevent the planner from revising the initial state in such a way that a character has two traits that are polar opposites. Consequently, an additional consistency check is performed every time an initial state revision is involved in the satisfaction of a personality recommendation. The planner checks the revised initial state for inconsistencies such as when a character is declared simultaneously to have a trait and to not have that same trait or when a character is declared simultaneously to have two traits that are polar opposites. A plan node with an inconsistent initial state is pruned from the search space. The consistency check does not affect the completeness of the planning algorithm; recall that any plan node that satisfies a personality recommendation on a plan step has a sibling plan node in which that recommendation is ignored.

6.5.2. Revising the Possible World Based on the Personality Model

Some extra provisions must be made when applying domain-independent support for a personal model to the search through the space of possible worlds. Personality recommendations are decision points. When ISR is combined with support for a personality model, how the recommendation decision points are addressed affects the possible world that the story is set in. Specifically, satisfying a recommendation with an initial state revision defines a subset of possible worlds in which a character has (or does not have) a particular personality trait. When there is no indeterminism about character personality traits, the planner non-deterministically chooses whether to satisfy a personality recommendation on a character action or to ignore the recommendation and leave the plan structure unmodified. Ignoring a recommendation signifies that the character is acting inconsistently. When there is indeterminism about character personality traits, the planner considers three distinct possibilities when repairing an open recommendation flaw on a plan step.

- The initial state is revised so that the personality trait in the initial state is determined to have the same value as is recommended and the open recommendation is satisfied by a causal link from the initial state to the plan step.
- The initial state is revised so that the personality trait in the initial state is determined to have the same value as is recommended and the open recommendation is *not* satisfied.
- The initial state is revised so that the personality trait in the initial state is determined to have the *opposite* value as is recommended and the recommendation is not satisfied.

In all cases a commitment is made to the character's personality. To ensure completeness of the algorithm, the planner considers all possible values that the personality trait can have. The planner also considers whether the recommendation is satisfied or ignored. If the personality trait is determined to be the opposite of the recommendation, as in the third case, then the recommendation cannot be satisfied by a causal link from the initial state. In the second case, a commitment is made to the personality of the character but the

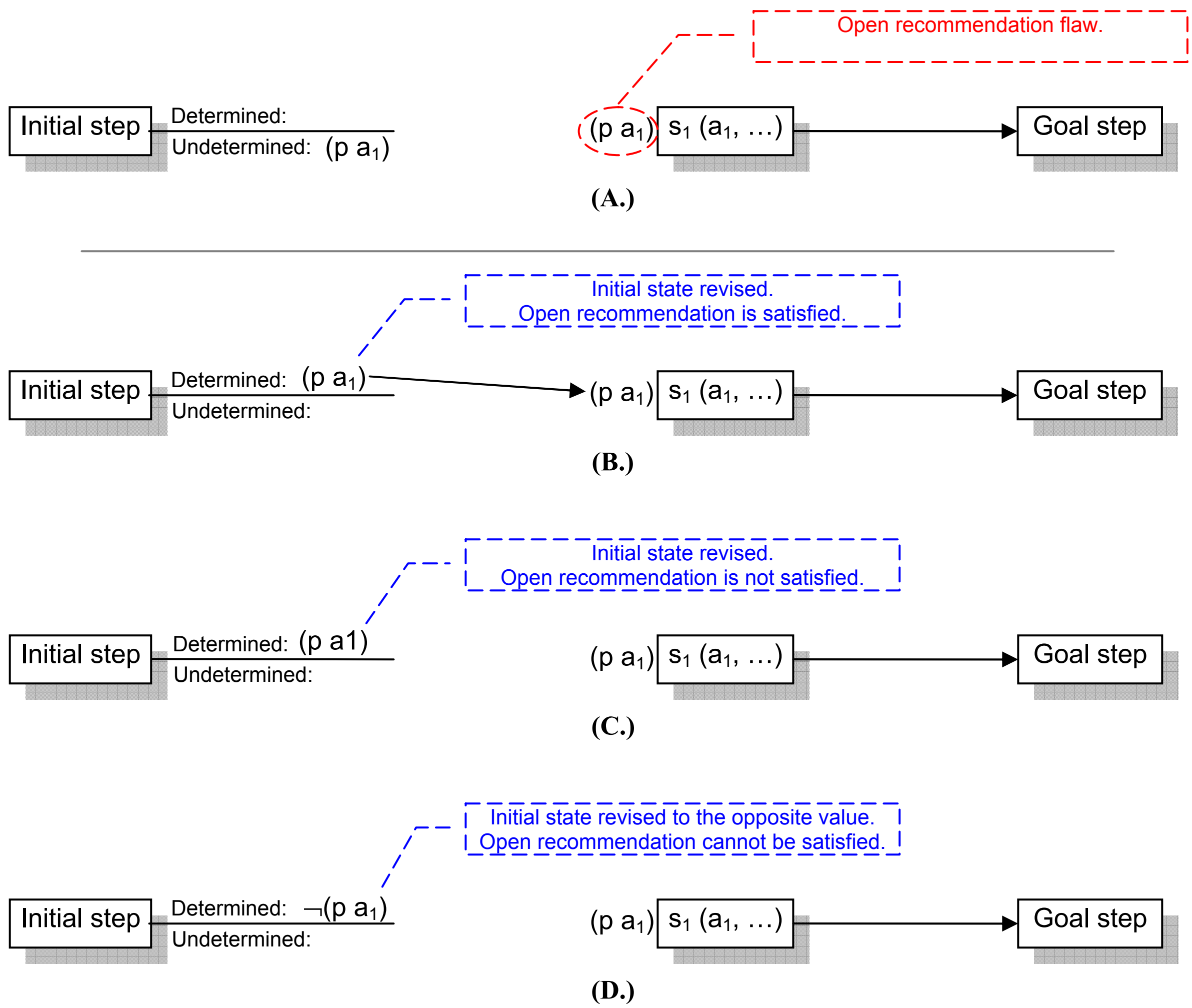

Figure 6.7. Ways in which open recommendation flaws are handled when a personality trait is undetermined.

recommendation in the plan step remains unsatisfied. This case allows for situations in which another plan step is inserted into the plan temporally prior to the plan step with the unsatisfied recommendation that changes the character's personality trait as an effect. Even though the personality trait of the character starts out the same as the recommendation, the character's personality can be changed by another plan step without creating a causal threat condition. The three cases are demonstrated in Figure 6.7. Figure 6.7.A. shows a fragment of a plan in which a plan step has an open recommendation flaw that has not yet been considered. Figures 6.7.B. through 6.7.D. show the three ways in which the open recommendation flaw is resolved through initial state revision. Initial state revision does not completely alter the way in which open recommendation flaws are resolved. If the recommendation condition does not unify with a sentence in the unknown set of the initial state, then the standard options for resolution are used, as discussed in Section 5.3.3.

6.6. Summary

The fabula planner is dependent on the way in which the human author defines the initial state of the world. The definition of the initial state of the world may make it impossible for a fabula planner to find a solution plan. It is also possible that the story planner find unwieldy or awkwardly structured stories because the initial state of the world is not conducive to the type of story that is generated. One way to rectify this situation is if the human author relinquishes some control over the definition of the initial state of the world to the story planner. Portions of the initial state of the world are left undetermined so that if one determination of the initial state is more conducive to the storytelling goals of the story planner, then the fabula planner can determine the actual value of certain sentences that describe the initial world state. If sentences are not relevant to the story being generated, they are left undetermined.

The sentences about the initial state of the world make up the undetermined set. Open conditions on actions in the fabula plan can unify with sentences in the undetermined set. When this happens, the sentence is determined to be known (true if the open condition is not negated or false otherwise), the initial state of the world is revised, and a causal link extends from the initial state to the action with the open condition. Revision of the initial state means that the value of that sentence about the world is never in question again. Initial state revision reduces the number of possible worlds that a story can be set in. To ensure that certain sentences can never be simultaneously true, sentences are grouped into mutual exclusivity sets. When one sentence in a mutual exclusivity set becomes known through initial state revision, then the other sentences in the set become known to have the opposite value.

If the human author leaves the personality traits of the story world characters undetermined, then the fabula planner implements the computational model of dramatic authoring.

Character traits are left unspecified by the human author so that, if a character needs a particular personality trait to perform a particular action, then the character can be determined to (a) have that trait and be acting "in character" or, (b) not to have that trait and be acting "out of character." The framework that supports personality relies on recommendations, in which action schemata suggest the traits that an acting character should have in order to perform that action. Recommendations that are satisfied by sentences in the undetermined set must be handled a bit differently than open conditions because recommendations do not need to be satisfied. The recommendation can be satisfied by revising the initial state to match the recommendation, the recommendation can be not satisfied by revising the initial state to the opposite of the recommendation, or the recommendation can be not satisfied while the initial state is revised to match the recommendation. Polar opposite traits are handled by making undetermined personality traits mutually exclusive in the initial state.

Chapter 7

Fabulist Architecture

Fabulist is the name of the system I have developed to generate stories that are both strong in plot coherence and strong in character believability. The name *Fabulist* refers to the system's role as the creator of a fabula. A fabula is the complete set of story world events in a story (Prince, 1987). The discussion of story planning and the planning algorithms described in Chapters 3 through 6, indeed, describe the details of generating a fabula. While the primary function of Fabulist is to generate a fabula, Fabulist is also capable of reasoning about how to tell a story and then actually telling the story. That is, Fabulist not only reasons about story on the level of fabula, but also on the level of sjužet and media.

Narratologists consider narrative as having three levels of interpretation: media, sjužet, and fabula. A narrative can be analyzed at any of those levels. Fabulist considers the structural interpretation of narrative as a literal model of narrative and views the process of narrative generation as the transformation of fabula into sjužet and of sjužet into media. The process does not necessarily model the way in which human authors create narrative. It is more likely that human authors work at all three levels simultaneously (Bal, 1997). A diagram of Fabulist's narrative generation process is shown in Figure 7.1. The model of the process of narrative generation is not to be confused with the model of dramatic authoring presented in Chapter 3; the model of dramatic authoring is a model of fabula generation and is thus merely one part of the overall process of generating narrative. Narrative is more than fabula alone; fabula is a sequence of events while narrative is the *recounting* of a sequence of events, implying the process of *telling*.

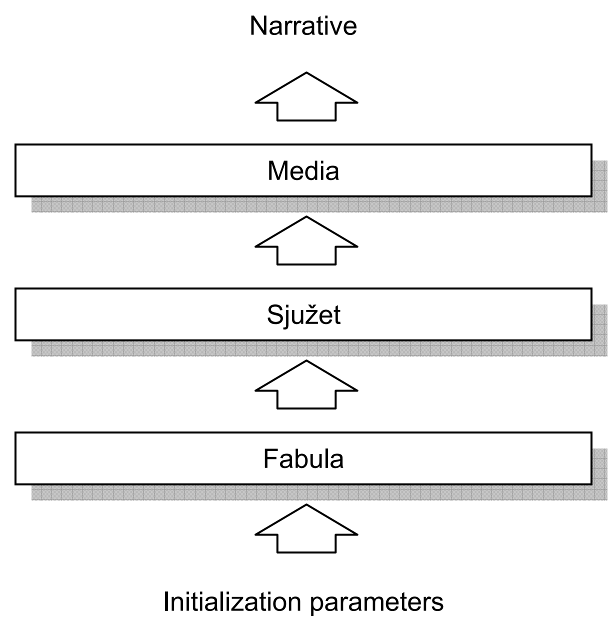

Figure 7.1. The narrative generation process used in Fabulist.

The architecture of Fabulist parallels the process model of narrative generation. Fabulist consists of three main modules, all connected to a domain knowledge-base as shown in Figure 7.2. The story planner module constructs a fabula plan as a data structure representing what a narrative is about. The story planner is initialized with parameters provided by the human author – the user. The fabula plan generated by the story planner is the input into the discourse planner module which finds a discourse plan consisting of communicative operations to be executed by a hypothetical narrator. The discourse plan is input into the media realizer which renders the discourse into a form that is appropriate for presentation to an audience, such as written or spoken text or cinematic visualization.

7.1. The Story Planner

The story planner module implements the computational model of dramatic authoring to generate a fabula in the form of a story plan. The fabula planner, encapsulated within the story planner module, is a combination of the IPOCL algorithm and the ISR planning algorithm and the personality model framework algorithm. These three algorithms modify different parts of the general partial-order planning algorithm and do not interact with each other when implemented together, except for ISR and the framework for a personality model

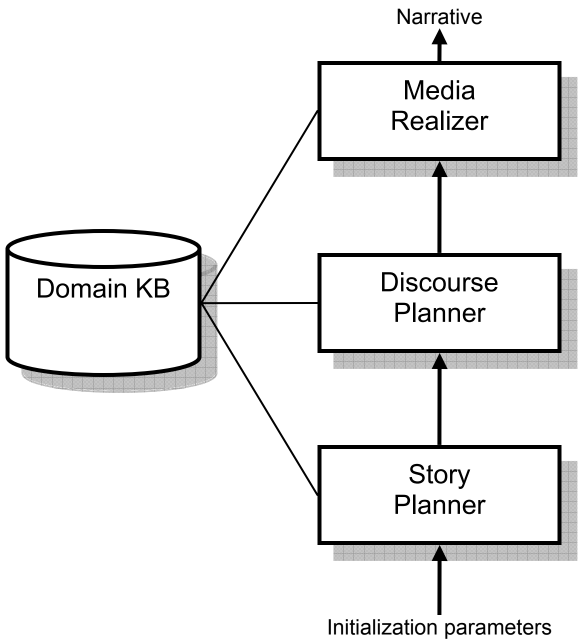

Figure 7.2. The Fabulist architecture.

whose interactions are described in Section 6.5. The fabula plan generated by the story planner module is not narrative. The fabula plan is merely a representation of change over time in the story world. For a fabula to be narrative, it must be narrated. The fabula is generated without considerations for discourse, according to the model of narrative generation. The following two subsections illustrate extensions to the basic IPOCL and ISR planning algorithms that help the story planner module support the computational model of dramatic authoring.

7.1.1. Author Goals

The outcome of the story being generated, given as a goal to be achieved by the fabula planner, is one part of ensuring that the generated story demonstrates the premise. The premise of a story, as proposed by Egri (see Chapter 3) is provided by the human author at initialization as (a) specification of story world characters, (b) the story world they inhabit, (c) the outcome of the story world, and (d) a conflict. The characters and the story world they inhabit are encoded as propositions that describe the initial state of the story world. The outcome is encoded as a goal – a set of atomic ground statements that partially describe the state of the story world once the story is complete. However, until now I have glossed over the story world planner's representation of the conflict inherent in the notion of premise. There are two ways in which the conflict of a premise can be captured in the domain of planning. The first way is to represent a conflict as one or more action operators that are instantiated in the initial (normally empty) plan in the plan space. This technique ensures that the story world characters will engage in conflict in a precise way. However, there is no guarantee that the outcome will rely on the effects of these actions. Any action that is not inserted into the plan through resolution of open conditions or open motivations (see Section 3.2.1) does not necessarily have to be on a causal chain terminating in the outcome of the story. By inserting actions into the story, the human author makes plot coherence uncertain.

The second way to represent the conflict of a premise is with *author goals*. Author goals are partial descriptions of the state of the story world that must be achieved at various points throughout the course of a story. Unlike the outcome which is represented by a goal, author goals must be achieved before the end of the story. Author goals are temporally ordered; if more than one author goal is specified by the human author, then there are temporal constraints that indicate that one partial world state must occur before the next. Author goals are conjunctions of atomic statements. Specifically, they are represented in the story plan as plan steps whose preconditions partially describe the state of the story world intended by the human author and whose effects are empty. Author goals are solved for as if they were ordinary steps with preconditions; an author goal is satisfied when all of its preconditions are satisfied, meaning that the partial description of the world state has been achieved at the particular time that the author goal occupies in the plan. Author goals are, of course, not executable operators, but place-holders. The only distinction between an author goal and the ordinary goal step, s_∞, is that author goals are not absolutely ordered after every other step in the plan.

The purpose of author goals allow the human author to exert a certain high-level degree of control over the direction the generated plot takes because the planning algorithm must solve for the author goals. This causes the planner to find only plans in which the author goal world states occur in the prescribed order. For example, Figure 7.3 shows a story plan with two author goals in addition to the outcome goal. The dashed lines represent temporal

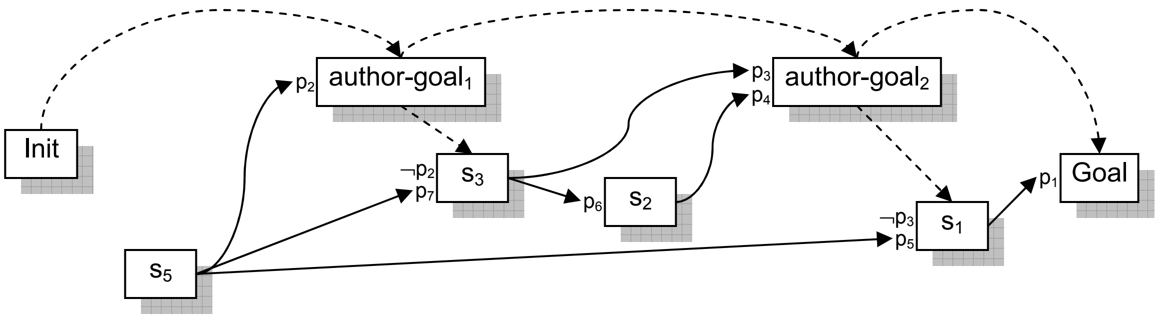

Figure 7.3. A story plan with author goals.

ordering constraints that are not otherwise captured by causal links (which imply a temporal ordering). *author-goal*$_1$ is satisfied by plan step s_5 and *author-goal*$_2$ is satisfied by steps s_3 and s_2. Step s_3 is necessarily ordered after *author-goal*$_1$ because it negates one of the conditions of the author goal. This illustrates that while the planner must find a plan that achieves each of the partial world state descriptions represented by author goals in order, the partial world states do not need to persist. Step s_3 could have been ordered before step s_5 except for the fact that it depends on one of the effects of s_5. Plan step s_1 is, likewise, ordered after *author-goal*$_2$ because it negates a condition of the author goal. In this case, however, step s_1 could have been ordered before step s_3; the choice is arbitrary.

The primary purpose of author goals is to enable the human author to specify the conflict of the premise. Instead of declaring the conflict to be made up of one or more exact actions, conflict is declared as one or more story world states. The expected story world states that are made into author goals may, for example, declare that certain propositions about the story world change value. For example, one author goal states that a frugal character is wealthy while another later author goal states that that same frugal character is poor. The human author may have it in mind that the frugal character becomes wealthy by not paying taxes and then becomes poor when the government comes after him. However, discretion as to exactly how the two author goal states, wealthy character and poor character, are achieved is given over entirely to the story planner. Heuristic functions can be used to favor certain legal scenarios over others.

Author goals serve a secondary purpose: to solve the reversal of fortune problem. Reversal of fortune is a situation where the human author intends that a proposition about the world changes value two or more times. Reversal of fortune is identified by Sgouros (1999) as the most important criterion for a successful story. For example, the human author may want a character to start out wealthy, become poor, and then become wealthy again. Reversal of fortune is a problem for all planning based story generation systems because the outcome cannot describe conflicting conditions. The goal of the story cannot declare both `(rich a₁)` and ¬`(rich a₁)` where a_1 is a story world character because any action that satisfied one condition will causally threaten the other condition making satisfaction of the goal impossible. Furthermore, if the goal state designates `(rich a₁)` and the initial state designates `(rich a₁)`, then more likely than not, the planner is going to satisfy the goal by maintaining the truth of the proposition throughout the story and there will never be a point in the story where ¬`(rich a₁)` is true. Author goals resolve the reversal of fortune problem by separating the conflicting conditions into separate goal states that must be achieved sequentially. Thus the same scenario as before – with the character that starts out rich, becomes poor, and then is restored to richness – can be represented with an initial state declaring `(rich a₁)`, a single author goal declaring ¬`(rich a₁)`, and the ordinary goal declaring `(rich a₁)`. It is up to the planner to figure out how the rich character becomes not rich and how that character then becomes rich again.

When the human author specifies author goals as initialization parameters into the story planner, story coherence of the generated story can no longer be assured by the planner because the user assumes some control over the evolution of the story world. If an author goal does not have some implied relevance to the outcome of the story, then the causal chains that lead to that author goal will be dead-end chains and dead-end chains imply that the story is not story coherent. For example, Figure 7.3 shows two actions, s_3 and s_2 that satisfy *author-goal*$_2$. Note that there is no path, following only causal links (solid arrows), from step s_2 to the goal. Whether or not the story plan in Figure 7.3 is story coherent or not depends on whether the audience perceives *author-goal*$_2$ as relevant to the outcome of the story. If *author-goal*$_2$ is perceived to be relevant to the outcome of the story, then there is an implied causal link from *author-goal*$_2$ to the goal state. If an author goal does not have implied relevance to the outcome of the story, then the story can be in one of two classes (See Figure 3.4):

- Plot coherent but not story coherent.
- Not plot coherent.

Which class the story belongs to depends on how the significance to which the audience attributes the events on the dead-end causal chain. If a dead-end causal chain has an event that is perceived by the audience to be a main event, then the generated story is not plot coherent.

7.1.2. Fabula Plan Ordering

The output of the fabula planner is a fabula plan that represents a story that has plot coherence and character believability. The fabula plan is partially ordered; there can be steps that have no explicit ordering constraints between them. Two steps are unordered relative to each other if neither step relies on the effects of the other step and if neither step has an effect that threatens the preconditions of the other step and if there is no intermediate step with which both steps are ordered relative to. A partially ordered plan represents a set of totally ordered plans (Knoblock, 1994). Some totally ordered plans in the set defined by a partially-ordered plan can be considered better than others. The classical assumption is that plan operators are indivisible and uninterruptible (Weld, 1994). However, just because two steps are unordered relative to one another does not mean they cannot be executed in parallel as long as there is no operational dependency between the two steps (Knoblock, 1994). Allen (1983) describes thirteen possible relationships between temporally significant events. Steps that are unordered with respect to each other (e.g. there are no ordering constraints and no intervening steps with which the steps have ordering constraints) can be in any one of the thirteen relationships.

Since the fabula planner does not consider any more than the least number of ordering commitments to ensure the fabula plan is sound, the complete fabula plan is input into a second process, a *final orderer*, that searches through all possible totally and partially ordered variations of the fabula plan that do not violate any ordering constraints for the best possible ordering. Ordering heuristic functions guide the search. Ordering heuristics evaluate a totally or partially ordered fabula plan based on the execution order of the plan steps. Like the planning heuristics used by the fabula planner, ordering heuristics can be categorized as domain-independent and domain-dependent. An example of a domain-independent heuristic is the suggestions that a motivating action should occur as close to the interval of intentionality it causes as possible (see Section 4.6). The current implementation of the final orderer does not consider the duration of actions and therefore can only distinguish between before, after, and simultaneous relationships. However extending the final orderer to support more sophisticated temporal relationships should be straightforward.

7.2. The Discourse Planner

The discourse planner module reasons about how best to tell the story represented by the story plan generated by the story planner. Different forms of media have different requirements for storytelling effectiveness, even though the story can remain the same. Since the discourse planner module is not a core contribution of my research, it is designed to encapsulate any number of discourse planners, depending on what the target medium is. If the target media is written or spoken text, then the discourse planner module can encapsulate an instance of the Longbow planner (Young, Moore, & Pollack, 1994), a sound and complete, decompositional partial-order planner for discourse generation. The decompositional nature of the Longbow planner enables the planner to capture the intentional structure of discourse in addition to its rhetorical structure (Moore & Paris, 1993). The discourse plans created by Longbow use operators that are similar to the relations used in Rhetorical Structure Theory (Mann & Thompson, 1987). The relations are persuasive in nature and consist of a nucleus concept that introduces a fact and a satellite concept that provides supporting evidence.

If the target medium is cinematic, then the discourse planner module can encapsulate an instance of a camera planner such as (Jhala, 2004) that is capable of describing camera placements and effects for capturing and presenting the actions of animated actors in a virtual world. The camera planner is also hierarchical and the operators implement film idioms that capture the conventions of visual storytelling. In the cinematic case, the discourse plan is similar to the fabula plan in that the discourse plan describes the actions that are components of the fabula plan. The discourse plan, however, has additional complexity because it includes camera actions that orient the audience's perspective in the virtual world. Camera actions are coordinated with the character actions to capture and convey the action in the virtual world to the audience in a meaningful way.

The purpose of the discourse planner module is to generate a media-specific discourse plan that describes what facts and in what order the facts about a particular domain are told to a hearer. The domain in this case is the fabula and the facts are the events that occur in the story as well as the relevant information required for a hearer to understand those events such as character and setting. The hearer in this case is the audience. The discourse planner is initialized with facts known by the speaker (the system) about the fabula and the single goal, `(knows hearer story)`. The goal is achieved by a sequence of plan steps are illocutionary acts performed by the speaker that operate on the knowledge of the hearer by asserting facts about the fabula. The discourse planner module is initialized with declarative statements about the fabula in the form of propositions about the knowledge of the speaker. The discourse planner represents the speaker agent as an authoritative expert on the events of the story. The fabula generated by the story planner module, however, is in the form of a plan – a directed, acyclical graph of plan steps and temporal and causal links. Before the fabula plan can be used by the discourse planner, it is first translated into declarative form by a straightforward routine.

7.3. The Media Realizer

The discourse planner module creates a discourse plan that describes how to best tell the story that was generated by the story planner. The discourse planner module, however, does not actually tell the story. That is, it does not execute the discourse plan. Execution of a discourse plan is to perform a series of primitive illocutionary acts that are aimed at altering the knowledge of the hearer. In this case, the hearer is the audience. Execution can mean one of several things depending on the target medium for storytelling. If the target medium is cinematic, then execution is the literal process of animated agents performing behaviors in a virtual world. If the target medium is oral or written communication, then execution means transforming the discourse plan into surface-level natural language text. The resulting natural language text can either be read by the audience or piped through a text-to-speech engine.

To realize the generated story in cinematic form, the discourse plan is treated like a script. The discourse plan contains a mixture of character actions and cinematic actions such as lighting (e.g. Seif El-Nasr & Horswill, 2003) and camera shots (e.g. Jhala, 2004). The media realizer is an execution engine attached to a 3D virtual world that can receive the discourse plan as input, such as the Mimesis system (Young et al., 2004). The character actions are passed off to animated agents in the order prescribed by the plan and the animated agents translate the discrete action specifications into animated behaviors. The virtual world execution engine implements procedural code packages for every action schema in the plan.

Figure 7.4. Tracking camera shot of a character running through the forest.

The procedural code packages execute in separate threads and detect their own success or failure conditions in order to coordinate parallel action execution. Some procedural code packages implement camera and lighting effects since the discourse plan contains specifications for these aspects of visual storytelling as well. Figure 7.4 (Jhala, 2004, fig. 22) shows a screenshot of a tracking camera shot of a character running. In this configuration, the discourse planner module is using the camera planner from (Jhala, 2004) and the Mimesis execution client. The animated agent that represents the character is executing a running action. The discrete operation in the story plan is procedurally realized in the execution engine as a culmination of many running steps and obstacle avoidance. The execution engine is simultaneously executing a discourse action that moves the camera to track the character's movement through the virtual world. Without camera control, the audience would be unaware of the movements of the characters through the virtual world.

To realize the generated story as oral or written communication, the discourse plan is translated into surface-level natural language text. The discourse plan achieves the communicative intentions of a hypothetical narrator through illocutionary actions that assert information about the events in the fabula plan. Execution of the discourse plan entails translating each primitive-level illocutionary assertion into text. Author (Callaway, 2000) is one system that generates prosaic natural language by generating scene descriptions and character dialogue from a narrative plan specification and even integrating multimedia content. Author segments the narrative into understandable chunks equivalent to paragraphs and sentences, makes lexical choices based on discourse history, and chooses from multiple points of view to generate a specification of the sentences that will be used to describe the story. The individual sentences are then planned and compiled in to functional descriptions that can be used by a surface realizer such as FUF (Elhadad, 1991). Author conflates the distinction between sjužet, as defined here, and media in that Author makes decisions about the point of view from which the story is told. However, a discourse planner is still required to transform the fabula into a "narrative stream" (Callaway, 2000), which is a data structure that describes the linear order in which assertions are to be made about the story. Many narratologists consider media realization part of discourse and therefore part of the sjužet.

While the Fabulist architecture supports various ways in which the story can be realized as media, the current implementation of Fabulist uses a simple template-matching routine to translate discourse actions into natural language. Primitive discourse actions are matched to templates and the parameters of the discourse actions are used to fill in the details of the template, resulting in one sentence per primitive discourse action. The primary contribution of my research is the story planner and the simple, template-based text realizer is a place-holder for a more sophisticated text realizer.

7.4. The Domain Knowledge-Base

The story planner, discourse planner, and, to some extent, the media realizer are domain independent, meaning that they make no assumptions about the story world or the genre of story being generated. The media realizer is not necessarily domain independent because it can be a custom designed virtual world with a domain specific graphical representation and customized character animations. However, this is not always the case. Due to the domain independent nature of the components of Fabulist, each component relies on a domain knowledge-base. The domain knowledge-base is a collection of domain specific information that enables each component to perform its duties. For example, the story planner module cannot generate a story unless it knows what characters, props, and locations exist in the story world and unless it knows what the valid operations through which those characters can manipulate their world are.

The story planner relies heavily on the domain knowledge-base for information about the story world. The domain knowledge-base contains the following pieces of information that are required by the story planner.

- Operator libraries for actions and happenings in the story world
- Planning heuristic functions used by the combined IPOCL, ISR, and personality framework algorithms.
- Ordering heuristic functions used by the final orderer.

- A personality model and any facts about personality and motivation that can be used by the planning heuristic functions.

The fabula planner and the final orderer utilize both domain-dependent and domain-independent heuristics. For convenience, both domain-dependent and domain-independent heuristics are contained in the domain knowledge-base since it is possible that a new genre of storytelling might need to override what was previously considered a domain-independent heuristic. The domain knowledge-base does not contain the definition of the initial world state, the goal state, or the author goals. These pieces of information, along with the domain knowledge-base, are direct initialization parameters for the story planner module.

The discourse planner module also relies on the domain knowledge-base, although depending on the specific discourse planner that is used in Fabulist, the domain knowledge-base may or may not need to store any domain-specific information related to discourse generation. For example, the current implementation of Fabulist uses the Longbow planner (Young, Moore, & Pollack, 1994) with a simple but robust set of domain-independent operators. The discourse planner module however does require some domain knowledge in order to transform the fabula plan generated by the story planner into a knowledge structure that can be used by Longbow, as described in Section 7.2. Specifically, the domain knowledge-base contains some simple rules about what data contained in the fabula plan the audience never needs to be told about. Depending on the story world domain operators (not the discourse planning operators) the fabula plan sometimes contains causal relationships between character actions that are necessary to ensure correctness of the fabula that can always be inferred by the audience. For example, for a character to be at some location in the story world implies that the character is not at any other location in the story world. Regardless of whether the discourse operator library is completely domain-independent or not, the discourse operator library is still contained within the domain knowledge-base to maintain modularity of the system. Discourse planning heuristics are also contained in the domain knowledge-base although these can likewise be either domain-dependent, domain-independent, or a mixture of both.

Finally, any domain-dependent information that the media realizer requires is part of the domain knowledge-base. The current implementation of the media realizer is a template-based text realizer and its templates are part of the domain knowledge-base. It is conceivable that the media realizer used in Fabulist be completely domain-dependent, in which case the knowledge required by the media realizer can be stored in the media realizer itself. It is also possible that the media realizer needs to know information about the initial state of the story world, in which case this information must be duplicated in the story planner module and the domain knowledge-base.

7.5 Implementation Notes

Fabulist's three components – the story planner, discourse planner, and media realizer – in addition to the domain knowledge base are implemented as part a single process. The output of each stage of the process of translating fabula into media is translated into input parameters from the next stage. The story planner, implementing the IPOCL, personality framework, and ISR algorithms was developed in Lisp by modifying the DPOCL planning algorithm (Young, Pollack, & Moore, 1994). The DPOCL planning algorithm is a decompositional partial-order planning algorithm that extends the UCPOP planner (Penberthy & Weld, 1992; Weld, 1994) with hierarchical operators. The Fabulist story planner disables decomposition (see Section 8.1.3. for a discussion of how decomposition can be integrated into intent-driven planning for character believability). The discourse planner used in versions of Fabulist built for this dissertation, including the empirical evaluation study in Chapter 9, uses an unmodified version of the Longbow planner (Young, Moore, & Pollack, 1994), also based on the DPOCL algorithm. The Longbow planner relies on decompositions to capture the levels of communicative intentionality. Specialized routines translate fabula plans into initialization parameters for the discourse planner and specialized heuristics guide the discourse plan generation process. The media realizer used in versions of Fabulist built for this dissertation uses a simple template-based matching routine to translate communicative actions in discourse plans directly into natural language.

Since the core contributions of this dissertation are implemented in the story planner, efficiency of the story planner requires some discussion. As mentioned in Section 4.7, the branching factor and search depth of the planning search space are increased by the IPOCL algorithm. The increase in the branching factor, especially, makes the generation of fabula plans impractical in real time. In the worst case, the branching factor of the search space is multiplied by (e + 1) where e is the highest number of effects of any action schemata in the story planner's action library. Unlike the simplified environments in which planners typically operate, the representation of story world states are larger and more complicated. Consequently, action schemata for story world actions have many preconditions and effects. Practical experience building even simple story world representations have resulted in search spaces where some intermediate nodes have hundreds of children (this is especially true when there are actions where more than one character is acting intentionally)! In Sections 4.6 and 5.3.5, I discuss how heuristics can be used to improve the decision-making ability of the story planner. The heuristics discussed in these sections can improve the average time to generate a fabula plan by focusing the planner on portions of the search space that are more likely to produce complete plans. However, none of the specialized heuristics were implemented. The domain-independent heuristics that were used focus on practical matters of plan efficiency such as length of plan, number of frames of commitment. The best-first performance of the story planner is not significantly better than that of a breadth-first search due to the high branching factor of the search space and the fact that most branches from any given intermediate node in the space were indistinguishable to the domain-independent heuristics actually implemented.

7.6. Summary

Narratologists distinguish between the fabula, sjužet, and media of a narrative. This suggests that, to some degree, the story content and discourse about the content can be reasoned about separately. The Fabulist architecture is based on the theoretical separation of fabula, sjužet, and fabula and defines three components: a fabula planner, a discourse planner, and a media realizer. The fabula planner uses a combination of the IPOCL, ISR, and personality framework algorithms to generate a sequence of character actions that has strong plot coherence and strong character believability. The discourse planner uses the fabula plan to generate a discourse plan that consists of communicative actions at several levels of abstraction. The media realizer renders the discourse plan into some presentation medium such as text or visual cinematography. The Fabulist architecture is designed to model the process of narrative generation (as opposed to the model of dramatic authoring which is relevant only to fabula construction) as a literal translation of fabula to sjužet to media.

In addition to the IPOCL, ISR, and personality framework algorithms, the fabula planner component also implements author goals. Author goals are temporally ordered partial descriptions of intermediate story world states that the human author can use to guide the generative process. Author goals force the fabula planner to find stories in which certain world states are achieved sometime between the beginning of the story and the end of the story. Author goals can be used to force the planner to incorporate conflict into the story and also to handle reversal of fortune situations. The fabula planner component additionally uses a final orderer routine to more completely order the actions in a fabula planner to reduce ambiguity of partial ordering. The final orderer does not necessarily result in a total ordering of the fabula plan.

Chapter 8

Limitations

The primary contribution of the research presented here is a story planner that can generate fabula plans with strong plot coherence and character believability. The story planner is an improvement over existing planning approaches to story generation because it reasons about character intentions in addition to the more standard author intentions. The bulk of limitations that I address are, consequently, limitations pertaining to the Fabulist story planner. Fabulist uses off-the-shelf discourse planning and media realization and the limitations of those systems are described elsewhere (e.g. Young & Moore, 1994; Moore & Pollack, 1992).

8.1. Story Planner Limitations

Most of the limitations of the story planner are direct consequences of using partial order planning as a model of story generation. While I believe that partial order planning makes a effective computational model of dramatic authoring, partial order planners are general problem solvers. Planners however, are general-purpose problem-solvers and are therefore primarily concerned with finding a sound and complete solution without necessarily reasoning about quality, aesthetics, or convention.

8.1.1. Conflict

As problem solvers, planning algorithms are designed to eliminate conflicts and failures. The fabula planner used by Fabulist, made up of the IPOCL planning algorithm, the ISR planning algorithm, and the framework for a personality model, does not have an explicit model for introducing conflict into the fabula plan. Conflict is an important aspect of good story. McKee (1997) suggests three levels of conflict: inner conflict in which a character struggles with his own mind, body, and emotions; personal conflict in which a protagonist struggles against those with whom he has personal relationships with; and extra-personal conflict in which a character struggles against society, the environment, or impersonal forces such as government. One form of conflict in story is when characters have inconsistent goals. As each character attempts to achieve his goal, he alters the state of the world in a way that threatens the ability of other characters to achieve their goals. Such conflict can manifest itself as causal threats in a POCL plan. Unfortunately, when conflict between characters manifests itself as causal threats, the planner attempts to eliminate the threats by ordering one character's actions before or after another character's actions or by backtracking to another branch in the search space that does not have a conflict.

While causal threats are an explicit form of conflict, conflict can also be implicit. The perceived struggle between characters is conflict, even though there may be no point in the story where one character's actions cause another character's actions to fail. Egri's (1960) notion of premise includes an implicit notion of conflict. The computational model of dramatic authoring, however, uses author goals as a model of conflict. That is, the planner is forced to consider plans with implicit conflict when the human author introduces author goals that force the planner to achieve certain intermediate states that suggest a struggle. Essentially, author goals recommend to the planner that the fabula plan take a certain form that it might otherwise have not considered because it is significantly more complex or involve significantly more causal threats that must be resolved. However, author goals place the burden of determining how conflict manifests itself in the story on the human author. Author goals *support* an implicit model of conflict – defined by the human author – but author goals are not a model of conflict.

A consequence of the way a fabula planner eliminates causal threats is that fabula plans are sound. That is, fabula plans do not fail to achieve the planning goal – the outcome. As a problem solver, the fabula planner is not able to consider failed actions. Character failure is a natural part of most stories and is especially important in comedy and tragedy (Charles, 2003). To insert an action into a fabula plan that fails means to leave one or more of an action's preconditions causally unsatisfied. From the perspective of the planner, actions that fail are contrary to the achievement of the planning goal and extraneous since other actions will have to be used to change the world state in a way that a failed action would have had it been successful. Similarly, the fabula planner cannot consider fabula plans in which character intentions are not successful. Consider the interval of intentionality of a character's frame of commitment to be a sub-plan that achieves the character's internal goal. The interval of intentionality is a DAG made up of actions performed by that character and causal links. Let there be some cutest that bisects the interval of intentionality such that all actions before the cut temporally precede the actions after the cut. The actions before the cut are executed but then, for whatever reason, the actions after the cut are not executed and the commitment to the internal character goal is abandoned. The story planner constructs plans by backward chaining from author and character goals meaning that the actions after any cut must be instantiated in the plan before the actions before a cut. The fabula planner simply cannot create a plan in which a character fails to achieve an intention without first considering how the character can successfully achieve the intention.

Not being able to construct fabula plans with failed actions or failed character intentions limits the types of stories that can be told by Fabulist. Fabulist cannot tell any story in which a character does not achieve its goal. This is problematic because story characters are often categorized as protagonist or antagonist and stories typically set up a world in which protagonists and antagonists have contradictory goals. In these stories, the protagonists triumph and the antagonists fail. Fabulist also cannot tell any story in which a character forms an intention and tries but fails to achieve the intention many times before finally succeeding. Stories should have a gap between expectations and results (McKee, 1997). That is, a protagonist should perform an action expecting a particular result but find that his action does not have the intended result, prompting a second attempt, and so on.

8.1.2. A Model of Story Structure

The story planner does not have an explicit model of story structure. The story planner utilizes explicit models of plot coherence and character believability because those are attributes of story that assumedly impact the success of a story. Plot coherence and character believability impacts the audience's comprehension of a story and support their ability to actively make inferences and predictions about the character's intentions and the direction of the plot. The success of a story is also dependent on the way that the audience relates to the story and the characters (McKee, 1997). The audience comes to expect a certain structure to story in the way it impacts them. Aristotle suggests that stories are structured in such a way that the tension the audience feels in response to the story changes in a predictable way over time as the story is told. Aristotle's dramatic arc, as shown in Figure 1.1, is one common model of story structure. In a story with more than one act, the dramatic arc of each successive act starts at a higher tension and the climaxes higher than the previous act such as in the dramatic arc used by the Façade system as shown in Figure 2.8.

The Fabulist story planner has a model of the process of story generation: the model of dramatic authoring. However, the story planner has no model of story structure, meaning that as it makes decisions about character actions that will be in the fabula, the decisions are made independent of the impact it has on the audience. Therefore, the story planner will not necessarily generate a story that has any coherent structure. Currently, the only way to get the story planner to generate stories with structure is to provide elaborate heuristic functions that map the actions, causal links, and temporal ordering constraints of a partially ordered plan into a vocabulary entailing tension or some other emotional or cognitive response the audience is expected to have to the story representation at any point in time. Defining heuristics that map fabula plan structure into emotional or cognitive impact is a difficult prospect. While the IPOCL algorithm expands the representation of a partially ordered plan, the plan representation is still relatively lean in that there are not many structural components, leaving a potential one-to-many relationship between plan representation and audience response. To complicate matters, fabula plans generated by the story planner are partially ordered, meaning the story planner makes the minimal number of commitments to the temporal ordering of steps needed to ensure plan soundness. The order in which actions execute can have an effect on the way in which the story impacts the audience. Finally, any heuristic function evaluates partially constructed plans after every decision point. It is unclear how to determine whether the audience's tension is rising or falling when there is no way to know whether actions in the partial plan will end up in the beginning, middle, or end of the complete solution.

Ultimately, however, heuristic functions can only suggest which of a set of alternative plans has the best structure. No alternative solution plan is guaranteed to have a desirable structure. Heuristics that could evaluate fabula plans with regard to story structure will be helpful to the current implementation of the Fabulist story planner because it will increase the likelihood that the story with the best structure is one of the first solutions plans found. However, heuristic functions do not increase the likelihood that a fabula plan with a good structure exists in the plan search space. The fabula plan representation and fabula planning algorithms should to be extended to consider structural decision points to ensure good story structure.

8.1.3. Decomposition of Actions

The current Fabulist story planner constructs fabula plans out of primitive actions. While reasoning at the level of primitive actions does not limit the stories that can be constructed by Fabulist, it does increase the number of non-deterministic decisions the story planner must make in order to create a complete fabula plan. That is, the planner must re-invent through the satisfaction of open conditions and the repair of causal threats situations that are familiar to the audience. Within a cultural context, the audience members have mental schemata corresponding to a wide variety of situations that story world characters may find themselves in. Some situations, such as ordering food at a restaurant, are familiar because audience members have constructed schemas from their direct experiences. Other situations, such as a bank robbery, are familiar because audience members have been told stories about how such situations. The story planner, however, constructs fabulas at the primitive level of actions. If a character has a goal of obtaining money, the planner determines that the character walks into a building and hands a slip of paper to a waiting employee who then walks to the back of the building, opens a vault, picks up some cash, returns, and hands the money over to the instigating character. All of this comes at great expense to the planner as it tries different alternative branches in the search space in which the character fails to obtain money. From the perspective of the audience, the entire sequence that was “invented” by the planner is instantly recognizable as a bank robbery.

To reason at the level of abstraction at which humans represent situations in the world is advantageous for two reasons. First, it decreases the amount of computation required to fabricate a coherent story plan. Second, the ability to represent situations as humans would means that it can fabricate story plans in which there are easily recognizable situations. To create a situation from the bottom up by piecing together primitive actions makes it possible for the planner to create a sequence of actions that is almost recognizable to the human audience but violates some expectations. Humans employ schemata to reduce cognitive load in familiar situations (Schank & Abelson, 1977; Rumelhart, 1980) and violating schemata results in an increased cognitive demand on the audience. Were the planner to reason at different levels of abstraction, the planner would be able to determine that a bank robbery is one way for a character to get money. Plan construction is performed as a top-down process of identifying abstract operators that are applicable and then decomposing those abstract operators into successively more primitive representations. The DPOCL planner (Young, Pollack, & Moore, 1994) does this by employing an operator library with operator schemata of varying levels of abstraction. Abstract operators are decomposed into more primitive operators by applying decomposition rules in a fashion similar to that used in NOAH (Sacerdoti, 1977). Decomposition rules are similar to the schemata that humans employ; they partially describe the operations that occur that make up the more abstract parent operator. Decomposition rules need only partially describe a situation in order to allow for some variability in how the abstract situation is achieved. Missing aspects of the decomposition rule that are filled in with a recursive invocation to the planning algorithm. DPOCL retains the option of piecing together a sequence from primitive actions regardless of whether an abstract operator exists. An example DPOCL plan showing the relationship between parent and children operations is shown in Figure 8.1 (Young, Moore, & Pollack, 1994, fig. 2). DPOCL is a sound and primitive complete[8] planning algorithm (Young, Pollack, & Moore, 1994).

[8] *Primitive complete* means that for every solution S to a planning problem α where S contains only primitive steps, the planner is guaranteed to produce a plan whose primitive steps are S.

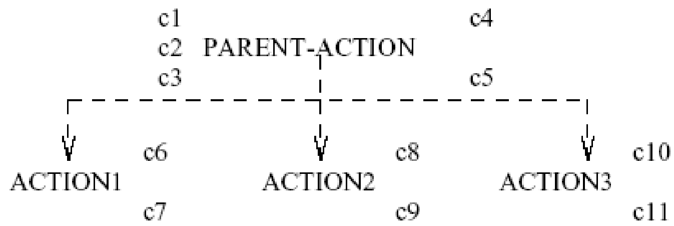

Figure 8.1. Schematic DPOCL plan illustrating parent/sub-plan relationship.

Like the IPOCL planning algorithm, the ISR planning algorithm, and the personality framework, the DPOCL planner is modifies the standard POCL algorithm. Decomposition can be integrated into the story planner module of Fabulist. However, it is likely that the decomposition of abstract operators is likely to interact with the way in which IPOCL resolves character intentionality. An abstract operator can represent an abstract *action*, meaning it represents a sequence of intentional behaviors. In this case the children actions are part of the same frame of commitment as the abstract action as long as the decomposed sequence does not reference any characters that are not referenced by the abstract parent action as acting intentionally. An example of an abstract action is `Rob-Bank`. The character assigned the role of robbing the bank is intentionally acting to achieve some effect of the operation. The children actions performed by the robber are part of the character's intention. However, if a bank teller is involved, the teller's actions are not part of the robber's frame of commitment. Abstract operators can describe *situations*, meaning that abstract operator does not ascribe intentionality to any character and the abstract operator is not part of any character's frame of commitment. An example of an abstract situation is `Bank-Robbery`. The decomposition may be the same as the example of the abstract action, but no assumptions are made about the intentionality of the character actions in the sub-plan. The planner must determine what the characters' intentions must be. Difficulties arise in determining how to assign intentionality to character actions that are inserted into the plan due to decomposition rules or are inserted to complete missing parts of partial decompositions.

Hierarchical planning as performed by the DPOCL algorithm can result in a speed-up in plan creation. The planner begins at a relatively high level of abstraction, requiring fewer operations because the high-level operations encapsulate sub-plans of lower level operations.

The high-level plan is a story coherent and, if combined with IPOCL, character believable. The high-level operators are decomposed using decomposition rules. Since lower level operations "fill in" the details of how a higher level operation is achieved, story coherence and character believability is preserved from one level to the next. If the decomposition rules are relatively complete, then the planner can insert entire sub-plans into the story plan. However, since decomposition rules are authored by a domain engineer, the planner cannot guarantee that decomposition rules include irrelevant sub-operations.

8.1.4. Relationship between Personality Traits and Actions

The framework for a domain-independent personality model relies on recommendations to determine if the actions performed by a particular character violate expectations. That is, a character's personality traits are encoded in the initial state of the world. Each action schema has personality recommendations that may or may not be satisfied by causal links. The personality framework assumes that actions performed by characters either demonstrate a trait or do not demonstrate a personality trait. However, personality is not necessarily always exemplified by discrete actions. Sometimes, the accumulated effect of a sequence of actions exemplifies a particular personality trait.

For example, suppose that the personality model has only two traits, good and evil. Actions in the story world recommend good or evil traits for the acting character or remain neutral. Character a_1 from country C_1 hires five other characters to carry five separate items into country C_2 where yet another character, a_2, assembles those items into a bomb. The act of hiring someone is neutral with respect to the personality model as is the act of carrying something. At this point in the story, character a_1 has not performed any action that can be considered good or evil. Yet, the effect of this sequence of actions is to smuggle a bomb into country C_2 which can be considered the behavior of an evil character. The audience will likely be able to understand the implications, although the planner will believe that character a_1 is acting neutrally. The action of character a_2 that assembles the bomb may or may not recommend an evil character. If the `Assemble` action is a general action, then the action schema will be neutral and the planner will not know that assembly of bombs is an evil action. Suppose the planner uses an `Assemble-Bomb` action that does in fact recommend that the acting character be evil. Either character a_2 is evil and acting accordingly or is not evil and acting "out of character." As the story unfolds, character a_2 uses the bomb to assassinate the tyrannical ruler of country C_2, saving millions of innocent people from genocide. The assassination action likewise recommends an evil character. However, the effect of assembling the bomb and assassinating the tyrant can be considered an act of goodness. The planner, however, only sees the way in which each individual action, taken in isolation, exemplifies or contradicts the personality definition of a character.

The heuristic functions discussed in Section 5.3.5 partially address the problem of the local relationship between personality traits and character actions. Heuristics evaluate the degree to which characters are acting consistently by counting the number of satisfied recommendations. However, heuristic functions can be written to recognize certain contexts in which the satisfaction of recommendations is not an accurate measure of consistency. The example given in Section 5.3.5 is how a character should be held accountable for inconsistent behavior when that character is being coerced. Social judgments, such as the effect of coercion or command structure in the military must be taken into account in order to determine appropriateness of character behavior (Mao & Gratch, 2004). In general, to correct the limited perception of the interaction between personality and action, the planner must completely model human social judgment and use it to evaluate aggregate actions of the story world characters from the perspective of the audience.

8.1.5. Support of Audience Recognition of Personality Traits

The framework supporting a domain-independent personality model uses the relationship between personality traits and character actions to weakly enforce consistency of character. Characters have personality descriptions in the initial state of the world and the actions they perform are matched up to that description as closely as possible. While the personality framework ensures that character actions are as consistent with personality descriptions as possible, the algorithm does not consider whether or not the audience shares the model of a character's personality with the planner. That is, the planner has a discrete description of a character to begin with but the audience must learn about a character by observing the actions the character performs. The assumption is that a character's personality will become clear through material cause to the audience over time. However, consistency is something learned through repetition. Each personality trait may only be associated with one or two character actions in the story plan. There is evidence that the character is acting consistently, but the audience may not be able to come to realize what traits a character has until too late.

Audience members need to learn about a character's traits before one can understand the role that that trait plays in how the story unfolds. One common way of resolving this dilemma is through *backstory* (McKee, 1997). The backstory of a narrative includes significant traits and events in a character's past that illustrate those aspects of a character that will become important to understanding the story later. Currently, the story planner used by Fabulist considers a plan to be flawed if it does not motivate why a character has an internal goal. Arguably, the planner should also consider a plan to be flawed if it does not illustrate a character's traits before they become significant to the outcome of the story. However, how to incorporate backstory into a story plan is not clear. Should the initial state of the story world, as defined in the planning problem (Definition 4.2) come before backstory elements? If so, then author goals can be used to make sure the planner includes actions early in the plan that illustrates character traits. However, this approach implies that the human author determines what character traits are to be important to the story ahead of time. This is contradictory to the model of dramatic authoring in Chapter 3 and precludes using initial state revision of character traits. If backstory must be created by the story planner and temporally occurs before the point in time that the initial state represents, how can the planner causally satisfy the preconditions of backstory events?

The Universe system utilizes a limited form of backstory as a way to "dropping hints" to the audience about how a particular situation might be resolved (Lebowitz, 1984). The Universe system requires characters to be as fully specified as possible before the planning process begins. A preprocessing phase simulates the life of each character in story world up to the point where the story begins. The lifecycle simulation is shown in the flow diagram in Figure 8.2 (Lebowitz, 1984, fig. 1). The information generated through the simulation process is stored in a data structure called a *person frame* for use during the story generation phase. If there is more than one alternative plot fragment that can be chosen by the planning process, the one that refers to a character's backstory the most is preferred (Lebowitz, 1985).

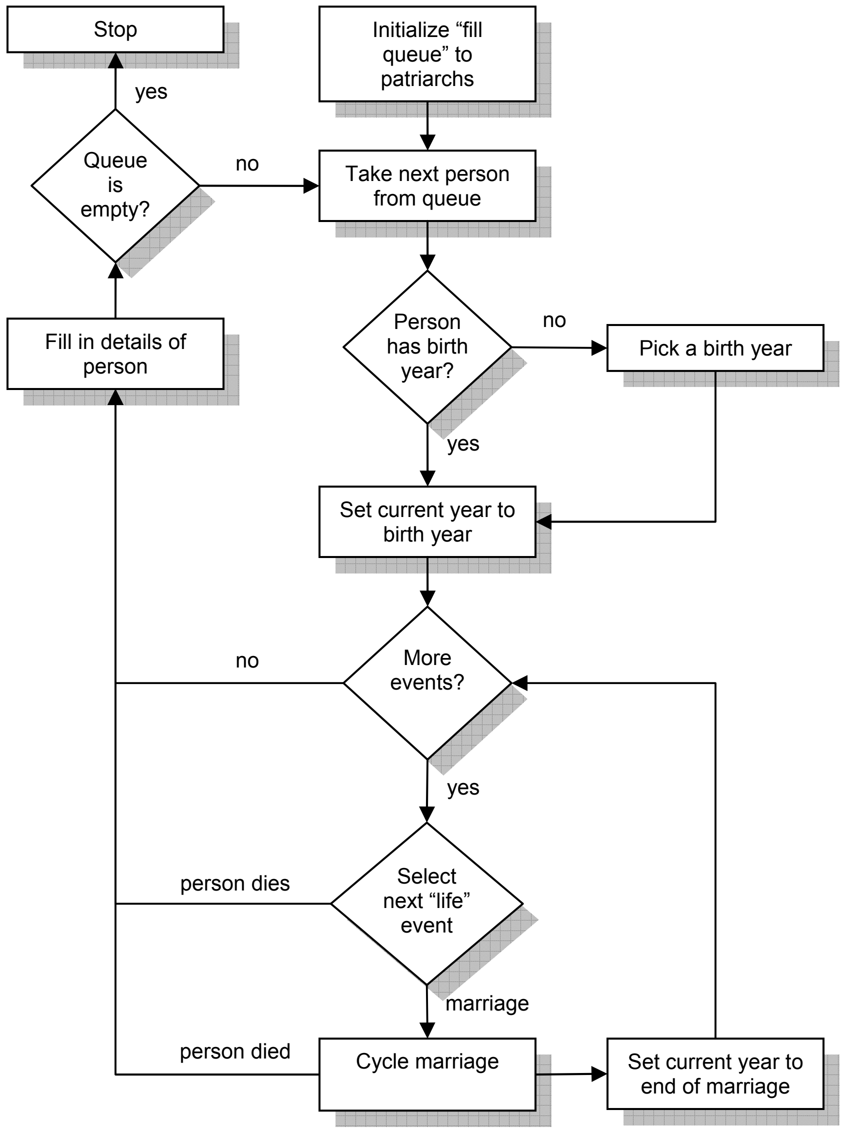

Figure 8.2. The Universe character “life cycle”.

Although backstory is generated, it is not necessarily used. Presumably, if it is used, the backstory events that are relevant will be told to the audience. The preprocessing approach, however, does not fit well into the model of dramatic authoring used in this work because it requires story world characters to be fully defined before story planning begins.

8.1.6. Designation of the Undetermined Initial State

Initial State Revision (ISR) is limited by the human author’s ability to foresee the need for flexibility. The ISR planning algorithm enables the planner to partially assume responsibility for determining the initial state of the story world. However, the planner can only do so when the human author has specified an indeterminate initial world state. While it is in the best interest of the human author to leave as much of the world state indeterminate as possible in order for the planner to produce a wider variety of solutions, the burden of specifying a viable world state on the human author is still great. The problem arises from the fact that the human author must explicitly think about what aspects of the story world the planner should have control over and what aspects of the story world should be absolute.

Furthermore, the human is limited in what can be left indeterminate. The atomic sentences in the undetermined set describe properties about the world or about objects and characters in the world. For example, the location of an object or a personality trait of a character would be left undetermined. What can be changed about the initial state of the world is the value of these properties. What cannot be changed about the initial state of the world is the existence of objects. That is, the planner cannot create or destroy objects or characters; the existence of any object in the story world is specified by the human author or is assumed to be false. The human author must specify every object in the initial state. If the human author is not certain whether the planner will require a second gun to construct a story plan, the human author must have the forethought to declare that a second gun exists but then leave all knowledge about the gun's properties undetermined. Even if the human author does this but the planner requires a third gun, then the planner still fails. The same situation can be applied to characters: the human author might need to specify extra characters to ensure that a solution can be found or that the solution is not awkward because it must work around a limited number of characters.

One way to reduce the severity of this limitation is to enable the planner to create objects and characters in the initial state of the story world as needed. Due to the closed world assumption, a planner assumes that any objects existence is false. Creating an object is equivalent to taking the existential statement about an object and changing its truth value from false to true. For example, the sentence `(gun g1)` is not specified as part of the $\mathcal{T}$ or $\mathcal{U}$ sets (see Chapter 6), implying that its truth value is false and the object does not exist. The statement of the object's existence may be false because the human author explicitly declares the object to not exist or because it is part of the universe of discourse that the human author has not considered. Of course, the planner has no way of knowing whether the object's non-existence is intended by the human author or not[9]. But if the planner were able to determine that the object's non-existence were an oversight by the human author, the planner could conceivably revise the initial world state so that the `(gun g1)` sentence were in $\mathcal{T}$ instead of $\mathcal{F}$. For every action that is instantiated in the plan, the planner could create a branch in the plan space in which a new object or character is created for every parameter of the action schema.

If the planner cannot find a solution plan using the objects and characters declared by the human author, it can fall back on alternative plans in which the initial state is revised to include objects and characters not considered by the human author. Constraints on object types for each parameter of each action schema can guide the planner to create objects of the appropriate type. Detailed descriptive sentences referring to newly created objects, such as the object's location, can be left undetermined and revised with the conventional ISR treatment. While the ability to dynamically create objects would significantly reduce the domain engineering burden on the human author, truth maintenance becomes complicated. If the existence of an object is changed from false to true and all sentences referring to the newly created object become undetermined (to provide the full flexibility offered by ISR planning), then some sentences about the object can become false again. There is then a danger that initial state revisions entailing the creation of new objects that occur later in the planning process can reverse earlier initial state revisions.

8.2. Architectural Limitations

While most of the limitations are present in the story planner module Fabulist, the architecture of Fabulist itself poses some limitations on the stories that can be told. The Fabulist architecture parallels the model of narrative generation which treats narrative generation as the process of transforming fabula into sjužet and transforming sjužet into media (see Figure 7.1). The segregation of the authoring process into three distinct stages of processing is artificial since human authors do not necessarily distinguish between fabula,

[9] Otherwise, the human author would be required to explicitly state every fact that is false in the world.

sjužet and medium during authoring. Furthermore, the segregation of authoring into distinct processes makes it impossible for higher layers to influence lower layers; sjužet cannot influence fabula and medium cannot influence sjužet or fabula. However, there are many instances in which the way a story is to be told has implications for the way in which the fabula is structured.

For example, suppose the story being generated is of the mystery genre. Narratives in the mystery genre typically follow a certain pattern in which a crime is committed at the beginning of the story but the information about who committed the crime is withheld from the audience. An explicit decision is made at the level of the sjužet to withhold information about the identity and circumstance in which the crime was committed. Were the discourse to be a literal rendition of the fabula, the criminal's identity would be immediately revealed. A fabula that is structured so that there is a clear delineation between the part of the story in which the crime is committed and the part in which a detective solves the crime will be more easily rendered into the genre-appropriate discourse structure. The structural considerations of the fabula plan may involve a minimization of causal links in the cut set that distinguishes one part from the other. Regardless of how the delineation is represented, a fabula planner that does not reason about discourse or does not receive input from a process that reasons about discourse will not necessarily make decisions about the structure of the fabula that is favorable for the *telling* of the story.

The same argument can be made for reasoning about media. There are some story events that can be told better in particular media and the strengths and weaknesses of the medium in which the story will be told should impact the contents of that are chosen by the story planner. Jhala (2004) also points out the limitations inherent in separating story generation and discourse generation into distinct processes.

8.3. Computational Complexity

As reported by Weld (1994), the computational complexity of POCL planners is $\mathrm{O}\left(cb^{n}\right)$ where

- *n* is the number of non-deterministic choices that must be made before a solution is obtained,
- *b* is the number of possibilities that need to be considered for each non-deterministic choice, and
- *c* is the time it takes to process a given node in the search space.

Because the average and worst case complexity of POCL planners in NP-Complete, story generation systems that rely on planning will never be applicable to systems that are required to generate narratives in real-time. As discussed in Section 4.7, the IPOCL algorithm increases both the branching factor and depth of the plan search space, although it is still NP-Complete. While the worst case complexity of is NP-Complete, there are techniques to reduce the average case complexity. Decomposition is one technique. Abstract operators are decomposed into more primitive sub-plans by applying decomposition rules. Decomposition rules declare sub-plans for achieving higher level abstract operators. The decomposition rule does not need to specify a complete sub-plan. If preconditions of operations in the sub-plan are not satisfied, a planner such as DPOCL (Young, Pollack, & Moore, 1994) will attempt to satisfy them. When decomposition rules specify complete sub-plans, these sub-plans are similar to task networks and the planning process becomes analogous to hierarchical task network (HTN) planners; the only decision points are those that determine which decomposition rule (if there is more than one applicable decomposition rule for a given abstract operator) to apply.

More sophisticated heuristic functions can also be used to achieve a speed-up in the time it takes to construct a narrative plan. While heuristics do not impact the computational complexity, they can be used to focus the planner on branches of the search space that are more likely to yield solutions. However, heuristics for story planning are highly domain-dependent and no such heuristic functions yet exist.

8.4. Summary

This chapter discusses some of the limitations of the Fabulist narrative generation system. Since the primary contribution of the research presented in this dissertation is a fabula planner that generates stories with strong plot coherence and strong character believability, the majority of the limitations revolve around the fabula planner component. The discourse planner and media realizer also have limitations that are mentioned in other research works.

One of the limitations of the story planner is that it cannot explicitly generate actions sequences with conflict. Conflict between the effects of actions is represented by causal threats which are resolved. Conflict between characters is not explicitly represented but would typically involve character performing actions that interfered with the goals of other characters. Since planners avoid additional structural complication caused by causal threats, mechanisms such as author goals must be introduced to induce the planner to consider more complex fabula structures that involve the implicit notion of conflict. Additionally, the story planner cannot generate stories in which actions fail or characters fail to achieve their internal goals. This limits the range of stories that Fabulist can tell. With regard to Fabulist's ability to tell stories, Fabulist does not have a model of plot structure and must rely on heuristic functions to favor plans that have a pattern of rising action and falling action. However, without an explicit model of story structure, there is no guarantee that the stories produced by the fabula planner will have any recognizable plot structure.

The nature of the representation of action in the fabula poses some limitations as well. The current state of the story planner only plans at the primitive level of action. However, there are many abstract situations that are familiar and even desirable to an audience. The story planner, unfortunately, would have to recreate these situations from scratch for such situations to be recognized by the audience. For the fabula planner to be able to reason about action at different levels of abstraction would mean more structured plots and more recognizable situations. The way in which character personality traits are associated with action schemata is also limited. Actions recommend personality traits. However, the audience does not necessarily infer a character's personality from the individual actions that the character performs, the audience also uses entire sequences and the purpose of those sequences to determine personality traits. Associating personality traits at the level of individual actions instead of at the level of sequences means that the audience can have a vastly different interpretation of a character than the one assumed by the system. Finally, enforcing character consistency with a personality model is a limited solution. The audience must learn about a character in order to recognize later on whether that character is acting consistently or not. However, the scope of the story plans that are generated by Fabulist does not necessarily go back far enough to establish character traits before those character traits become relevant to the way in which the story outcome is achieved.

ISR planning is limited in that it does not necessarily reduce the amount of effort required for the human author to specify the initial state of the world. In fact, ISR planning can make the description of the initial state of the world more complicated by insisting that the human author think about what needs to be left undetermined and what does not.

There are limitations of Fabulist that are the result of assumptions built into the architecture. Specifically, the Fabulist architecture assumes that story content and discourse are completely separate. However, the story content that is constructed by the story planner can have implications for how the story is told.

Chapter 9

Evaluation

The Fabulist system models the process of narrative generation as the transformation of fabula into sjužet and media. While the core contribution of this work to the field of computational story generation is the construction of a fabula plan by the story planner, Fabulist is capable of taking a fabula plan and presenting it in an arbitrary medium. The most common configuration of Fabulist is to use a text realizer that outputs a natural language text rendition of the generated story. The goal of the Fabulist system – the story planner module in particular – to generate stories that have strong plot coherence and strong character believability. However, plot coherence and character believability are not computational attributes of a story generator, but perceived attributes of stories themselves. That is, whether or not a story generated by Fabulist has plot coherence or character believability depends on whether an audience perceives those qualities in the story. Empirical evaluation of something as subjective as audience perception of plot coherence and character believability in a story is challenging for the following reasons.

- While the concepts of plot coherence and character believability are universal, audience members do not necessary think of stories in those terms or even share a common definition of those terms. Therefore, the audience cannot be directly polled with regard to plot coherence and character believability after being presented with a story.
- Plot coherence and character believability are attributes of story and are therefore attributes associated with the fabula plans generated by the story planner module.

However, the audience cannot be presented with a fabula plan data structure. Instead, the audience must be presented with narrative form rendered into some communicative medium. There is no guarantee that the transformation of fabula into discourse alters the audience's perception of plot coherence and character believability.

- Plot coherence and character believability are attributes of story that ultimately impact the audience's comprehension of a story. Subjective opinion polls cannot control for unintended artifacts such as the quality of natural language or graphics rendering that can affect an audiences like or dislike of a story.

In light of these challenges, I believe that the best way to empirically evaluate whether an audience perceives a story generated by Fabulist to have plot coherence and character believability is to indirectly evaluate the cognitive representations formed by audience members during the presentation of a story. The cognitive representation of a story held by an audience member is not directly accessible, but the structure of the cognitive representations of that story will affect the performance of that audience member on certain tasks related to the story, such as recall (van den Broek, 1988) or question-answering (Graesser, Lang, & Roberts, 1991). Furthermore, the indirect evaluation of cognitive representations of story in an audience member's mind is not subjective. There are various theories of how stories are represented cognitively (e.g. Rumelhart, 1975; Thorndyke, 1977; Schank & Abelson, 1977; Black & Bower, 1980; Wilensky, 1982; Lehnert, 1982; Trabasso & Sperry, 1985; Graesser, Lang, & Roberts, 1991) and how those structures are related to story comprehension.

Although in Section 3.1, I argue that plot and character are interrelated, plot coherence and character believability are orthogonal. That is, I believe that it is possible for a story to have plot coherence without character believability or to have character believability without plot coherence. The orthogonal nature of these story attributes is the basis for the distinction between author-centric story generation systems and character-centric story generation systems (see Section 2.1). Furthermore, the orthogonal nature implies that the concepts of plot coherence and character believability can be studied separately. I have designed and run two distinct evaluation studies to determine empirically whether stories generated by Fabulist have plot coherence and story generation. The first evaluation study addresses plot coherence in Fabulist-generated stories. The second evaluation study addresses character believability in Fabulist-generated stories.

9.1. Plot Coherence Evaluation

In order to determine whether stories generated by Fabulist have plot coherence, I need to ascertain whether the audience can identify the ways in which the events of a story are related to the outcome of the story. I make the assertion in Section 3.2.1 that the story planner generates stories that are story coherent and that story coherence is a limited form of plot coherence. Therefore, the purpose of the evaluation study presented here is to demonstrate with statistical significance that stories generated by Fabulist are story coherent. The computational model of story coherence in Section 3.2.1 makes the case that the causal links represented in the fabula plan guarantee story coherence because each action in the fabula plan is causally necessary and sufficient for the outcome of the story to be achieved.

Trabasso and Sperry (1985) determined that there is strong correlation between causal relatedness and the importance of story events. Two events in a story are causally related if the latter event could not occur unless the earlier event had not occurred. Causal relationships were established by applying a counterfactual argument of the form: If not A then not B. That is, an event A is said to be necessary to event B if it the case that had A not occurred then B would not have occurred. Brown and Smiley (1977) measured the perceived importance of events in six stories. Trabasso and Sperry derived causal networks for the same six stories using their counterfactual technique and all causal chains that terminated in the outcome of the story were identified. The importance rating of events from (Brown & Smiley, 1977) were compared to the causal networks for the stories to determine that importance of an event was strongly correlated with the number of causal connections the event had and whether or not the event was on a causal chain that terminated in the outcome of the story or a dead-end chain (Trabasso & Sperry, 1985).

The perceived importance of an event in a story is correlated with whether or not that event is part of a causal chain or a dead-end. Events that are part of causal chains, and therefore contribute significantly to the outcome of a story, are rated more important than events that do not contribute significantly to the outcome of a story. By applying the findings of (Trabasso & Sperry, 1985), an event in a story can be identified as being on a causal chain or a dead-end by determining its perceived importance. A story coherent story has no dead-ends. Given this interpretation, the purpose of this evaluation study is to empirically establish whether a story generated by Fabulist is perceived to be story coherent. A story will be perceived to be story coherent if all of the events in the fabula are deemed relevant to the outcome of the story. That is, a story will be perceived to be story coherent if none of the events in the fabula are deemed to be not important.

The fabula planner used by Fabulist generates fabula plans that explicitly represent all causal relationships between character actions. A fabula plan is a causal network in which all paths through the causal network terminate in the goal state of the plan which defines the outcome of the story. Plot coherence, however, is a perceived attribute of story. That is, a story is not plot coherent unless the audience believes it to be plot coherent. The model of plot coherence used by the fabula planner in Fabulist is based on the fact that an audience's perception of plot coherence in a story is related to the causal connectedness of the events in the story. The fabula planner constructs fabula plans such that all actions are on causal chains terminating in the story goal. Consequently, the fabula planner cannot generate a story plan that has actions that are not causally relevant to the outcome of the story. Given the importance of causal chains to the audience's perception, if a story were to have actions that were part of a dead-end causal chain, then the audience's perception of plot coherence should diminish. To test the audience's perception of story coherence, I take a narrative generated by Fabulist and insert events that are hand-authored by me so that there are dead-ends in the causal network. The hand-authored actions were carefully chosen so that they cannot be mistaken as part of a causal chain that terminates in the outcome of the story. If the narrative generated by Fabulist (without the hand-authored parts) is story coherent, then a reader will identify the Fabulist-generated actions as important and the hand-authored actions as unimportant. For a reader to fail to rate Fabulist-generated actions as important and hand-authored actions as unimportant indicates that the reader does not comprehend the causal significance of actions in the fabula and that the Fabulist-generated story is not story coherent. The following hypothesis is being tested:

> **Hypothesis 1.** *If a narrative generated by Fabulist is story coherent, then the mean importance rating of the events generated by the fabula planner will be higher than the mean importance rating of the hand-authored events.*

9.1.1. Method

To determine whether subjects perceived story coherence in stories generated by Fabulist, I used a version of Fabulist to generate a narrative. Subjects were asked to read the story and rate the importance of each sentence in the narrative. The implementation of Fabulist used for this study had a story planner implementing the IPOCL algorithm, a discourse planner based on the Longbow planner (Young, Moore, & Pollack, 1994), and a template-based text realizer. Fabulist was initialized with a description of a story world, a goal state to be achieved in the story world, and an action library that had a sufficient number of actions to successfully generate a story plan. The initialization parameters are described in Appendix B.

Fabulist was used to generate a story of significant length, consisting of 11 events. The discourse planner module encapsulated an instance of the Longbow planner (Young, Moore, & Pollack, 1994) with a simplified discourse operator library. The media realizer used a simple template-matching approach to render the discourse plan operations into natural language. The nature of the text realizer was such that one sentence was generated for each statement about the story world and the events that occur in the story world.

If Fabulist stories are story coherent, the fabula planner used by Fabulist cannot generate stories with dead-ends. In order to establish a baseline against which to evaluate the importance ranking of story events on causal chains, four additional events were hand-authored by me and added to the story to create a story with dead-ends. The story used in the evaluation is shown in Figure 9.1. The portions of text in italics were hand-authored while the remaining text was generated by Fabulist. The hand-authored portions of the text were written in the same stilted style as the generated portions of the story to ensure that the subjects would be unable to determine that there were two different authors. There are two hand-authored sequences: one involves the protagonists helping out a peasant and the other involves a jester proclaiming the marriage of the King and Jasmine. These sequences were chosen because they are plausible additions to the basic plot of the story. The protagonist's charitable actions illustrate the virtues of the protagonist and the proclamation of the marriage is a natural evolution of the main events of the story. A graphical representation of the fabula plan generated by the story planner (before being transformed into a discourse plan and text realization) is shown in Figure 9.2. The plan shows the events and frames of commitment for the characters, although many conditions and causal links are omitted for simplicity.

Trabasso and Sperry (1985) have demonstrated that events that are on causal chains that terminate with the outcome of the story are perceived to be more important than events that are on dead-ends. If the hypothesis is true and the generated events are perceived to be more important than the hand-authored events, then subjects will have clearly been able to identify the events generated by the story planner as being causally relevant to the outcome of the story.

The evaluation was set up as a questionnaire in which subjects read the story text containing computer generated and hand-authored components of the story. The subjects were not told that the story was generated by a computer program. After the text of the story, the sentences were listed in a random order and subjects were asked to rank each sentence based on their perception of importance. The order of the sentences was randomized in the questionnaire portion in order to control for the possibility that the flow of one sentence to another might affect the subject's perception of importance. Additionally, randomization ensured that the subjects would have to rely on the cognitive representation of the story that was formed when reading through the story text. Sentences in the randomized list were rated on a four-point Likert scale ranging from 1 to 4 where 1 meant "not at all important" and 4 meant "very important". The subjects were not given a definition of "important" and were left to make their own determination of what it meant for a sentence to be important. Subjects were asked to read the story text completely at least once before proceeding to the ratings task. Subjects were allowed to refer back to the original text at any time during the rating task. In addition to the importance ratings, I also asked subjects to estimate the number of times they had to refer back to the original text while performing the rating task.

Sixteen undergraduate students in the Computer Science program at North Carolina State University participated in the study. All subjects were enrolled in the course, *Game Design and Development*, and were compensated for their time with five extra credit points on their final grade in the course. Since I was only interested in whether events were part of a causal chain or a dead-end, all sentences that did not directly correspond to an event in the fabula plan were disregarded in the results analysis.

There is a woman named Jasmine. There is a king named Mamoud. This is a story about how King Mamoud becomes married to Jasmine.

There is a hoard of gold. There is a dragon. *The dragon has the hoard of gold.* There is a magic lamp. The dragon has the magic lamp. There is a magic genie. The genie is confined within the magic lamp.

King Mamoud is not married. Jasmine is very beautiful. King Mamoud sees Jasmine and instantly falls in love with her. King Mamoud wants to marry Jasmine. There is a brave knight named Aladdin. Aladdin is loyal to the death to King Mamoud. King Mamoud orders Aladdin to get the magic lamp for him. Aladdin wants King Mamoud to have the magic lamp. Aladdin travels from the castle to the mountains. Aladdin slays the dragon. The dragon is dead. Aladdin takes the magic lamp from the dead body of the dragon. *There is a peasant named Ali. Ali is in the mountains. Ali is very poor. Ali begs Aladdin for some money. Aladdin is very generous. Aladdin wants to help Ali. Aladdin takes the hoard of gold from the dead body of the dragon. Aladdin gives the hoard of gold to Ali.* Aladdin travels from the mountains to the castle. Aladdin hands the magic lamp to King Mamoud. The genie is in the magic lamp. King Mamoud rubs the magic lamp and summons the genie out of it. The genie is not confined within the magic lamp. King Mamoud controls the genie with the magic lamp. King Mamoud uses the magic lamp to command the genie to make Jasmine love him. The genie wants Jasmine to be in love with King Mamoud. The genie casts a spell on Jasmine making her fall in love with King Mamoud. Jasmine is madly in love with King Mamoud. Jasmine wants to marry King Mamoud. *There is a court jester. The court jester heard about the wedding. The court jester wanted everyone to know about the wedding. There is a town near the castle. The court jester went to the town. The court jester proclaimed the marriage of King Mamoud to Jasmine.* The King Mamoud and Jasmine wed in an extravagant ceremony.

King Mamoud and Jasmine are married. The end.

Figure 9.1. Narrative text used in the plot coherence evaluation study.

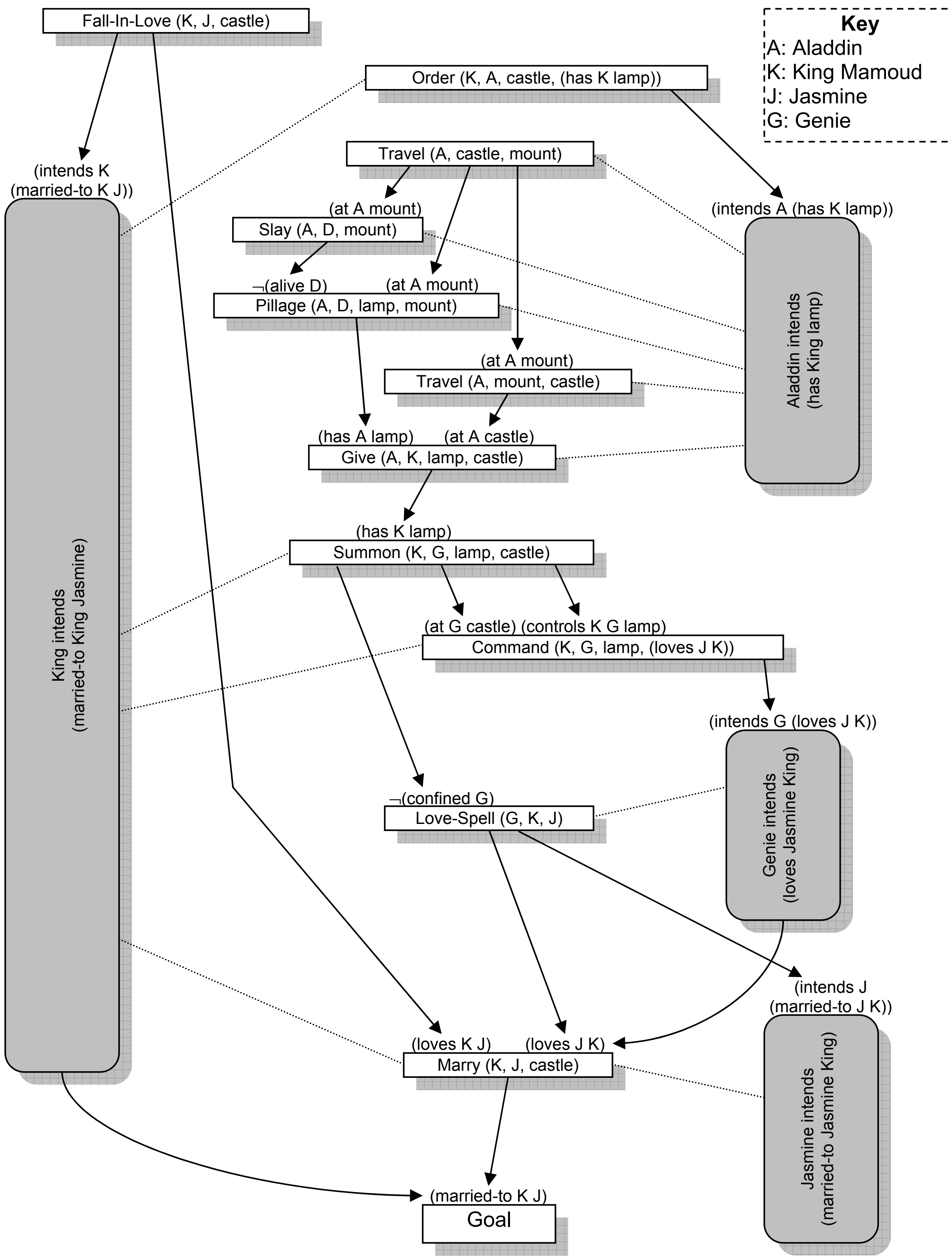

Figure 9.2. Fabula plan representation of the story in the plot coherence evaluation study.

9.1.2. Results

The results of the evaluation are shown in Table 9.1. A standard one-tail t-test was used to compare the mean importance rating of the generated events to the mean importance rating of the hand-authored events. The result of the t-test with 15 degrees of freedom yields t = 8.2570 ($p < 0.0005$). The hand-authored portions of the story are causal dead-ends while the generated events are on valid causal chains. The fact that the subjects were able to unequivocally distinguish the hand-authored events from the Fabulist generated events strongly suggests that, in the light of the study by Trabasso and Sperry, all story events generated by Fabulist are perceived to be on causal chains that terminate with the outcome of the story. Since story coherent stories do not have dead-ends, the evidence suggests that stories generated by Fabulist are story coherent, supporting the hypothesis. Since all story coherent stories are plot coherent as well, stories generated by Fabulist are plot coherent.

The average number of times a subject had to refer back to the original text while rating the randomized list of sentences was 3.06 times with the median being 2. There are a total of 49 sentences in the story text. The relatively small number of referrals to the original text given the relatively large number of sentences suggests that subjects relied heavily on their memory of the story from their first reading.

9.1.3. Discussion

The fabula plans generated by the story planner are directed acyclical graphs consisting primarily of plan steps, temporal ordering constraints, and causal links. The causal links create a causal network not dissimilar to the causal networks identified by Trabasso and Sperry (1985). The audience is not given the graph to interpret, however. Instead, the graph is transformed into a discourse plan and then into text. The explicit causal connectivity becomes implicit in the telling, although it is quite possible that the causal connectivity between events is completely lost. As active readers, the subjects of the study reconstruct

Table 9.1. Results of the plot coherence evaluation study.

	Mean importance rating	Standard deviation
Generated events	3.0966	0.2538
Hand-authored events	1.6406	0.6581

their own interpretations of the fabula (Thorndyke, 1977; Thorndyke & Yekovich, 1980; Black & Bower, 1980; Gerrig, 1993). The results of this study indicate that the subjects reconstruct the causal network in such a way that the events generated by the story planner are considered to be more meaningful than the events that were hand-authored. The hypothesis states that stories generated by Fabulist are story coherent. More specifically, this means that fabulas generated by the story planner are story coherent. The fact that all events generated by the story planner were perceived to be important suggests that all events generated by the story planner are causally relevant to the outcome of the story.

The standard deviation of the ratings of the generated events was lower than the standard deviation of the ratings of the hand-authored events. This suggests that subjects were much more certain of their ratings of the generated events than of the hand-authored events. One explanation for the difference is that subjects were free to determine their own definition of what it means for a sentence to be "important." The results indicate that subjects were in very close agreement when an event was important, but were in less agreement about events that were not as important. Subjects did not necessarily rate a sentence as unimportant if it was not causally relevant. It is possible that some subjects saw the events demonstrating the protagonist's chartable nature as important because the protagonist was essentially performing unselfish acts of heroism throughout the story for the benefit of others (such as the King), even though there was no causal relevance of the charitable events to the outcome of the story. Likewise, it is possible that some subjects saw the events of proclaiming the marriage as important because of expectations about the genre (e.g. Kings in fairy tales have extravagant weddings that are proclaimed throughout the land). In addition to the text base of the story, readers also rely on *generic knowledge structures* in long term memory that encode patterns and expectations derived from experience reading other, similar texts (Graesser, Lang, & Roberts, 1991). The study did not control for generic knowledge structures.

9.2. Character Believability Evaluation

In order to determine whether stories generated by Fabulist have plot coherence, I need to ascertain whether the audience can identify the actions performed by story world characters as being driven by their internal goals, desires, and traits. Whereas story coherence can be directly modeled by the causal representations in a POCL plan, achieving character believability in fabula plans requires much more extensive modifications to the POCL algorithm. The story planner used by Fabulist implements the IPOCL algorithm and the personality framework algorithm to establish intentionality of character actions and consistency of character actions, respectively. Computationally, I assert that a fabula plan has character believability if the following conditions are true.

- All character actions are associated with a frame of commitment and that frame of commitment is causally motivated by a domain-level action
- All character actions have all their personality recommendations satisfied.

The algorithm, however, allows for the possibility that a character action has unsatisfied personality recommendations because it is possible, and perhaps even desirable in certain circumstances, that a story world character act "out of character". Furthermore, for an audience to perceive that a character is acting consistently, the audience must become aware of a pattern of character decision-making that can be attributed to a particular trait. Not only might the audience use a different model of personality with which to evaluate a character, but the story must be long enough and have enough situations that demonstrate the way a trait affects a character's behavior for the audience to gradually build a mental model of that character. Consistent, trait-driven character behavior is, consequently, extremely difficult to evaluate. Since the story planner algorithm is NP-Complete, story generation becomes exponentially harder as the story length (e.g. the number of non-deterministic choices about character actions that must be made) grows. For these reasons, I have limited my evaluation of character believability in Fabulist-generated stories to be an evaluation of intentionality of character behavior in Fabulist-generated stories.

To empirically test a reader's perception of character intentionality, I based my evaluation study on the QUEST model of question answering (Graesser, Lang, & Roberts, 1991). The QUEST model accounts for the goodness-of-answer (GOA) judgments for questions asked about passages of prose. The purpose of the QUEST model is to show that people build cognitive representations of stories they read that capture certain relationships between events in a story and the perceived goals of the characters in the story (Graesser, Lang, & Roberts, 1991). The cognitive representation of the relations in the story is queried when the reader answers questions about the story. The types of questions supported by the QUEST model are: why, how, when, enablement, and consequence. For example, the story in Figure 9.3 (Graesser, Lang, & Roberts, 1991, table 1) has a corresponding QUEST knowledge structure shown in Figure 9.4 (Graesser, Lang, & Roberts, 1991, fig. 1). There are two types of nodes in a QUEST knowledge structure: event nodes, which correspond to occurrences in the story world, and goal nodes, which correspond to goals that characters have. The links in Figure 9.4 represent the different types of relationships between events and character goals.

- Consequence (C): The terminal event node is a consequence of the initiating event node.
- Reason (R): The initiating goal node is the reason for the terminal event node.
- Initiate (I): The initiating event node initiates the terminal goal node.
- Outcome (O): The terminal event node is the outcome of the initiating goal node.
- Implies (Im): The initiating event node implies the terminal event node.

Graesser, Lang, and Roberts (1991) illustrate the QUEST model of question answering with the following question pertaining to the story in Figure 9.3: "Why did the daughters stay in the woods too long?" The question is a query about node 5 in Figure 9.4. There are many possible answers, some of which are shown below.

A. Because the daughters forgot the time (node 4).

B. Because the dragon kidnapped the daughters (node 7).

> Once there was a Czar who had three lovely daughters. One day the three daughters went walking in the woods. They were enjoying themselves so much that they forgot the time and stayed too long. A dragon kidnapped the three daughters. As they were being dragged off, they cried for help. Three heroes heard the cries and set off to rescue the daughters. The heroes came and fought the dragon and rescued the maidens. Then the heroes returned the daughters to their palace. When the Czar heard of the rescue, he rewarded the heroes.

Figure 9.3. An example story used to test the QUEST model of question answering.

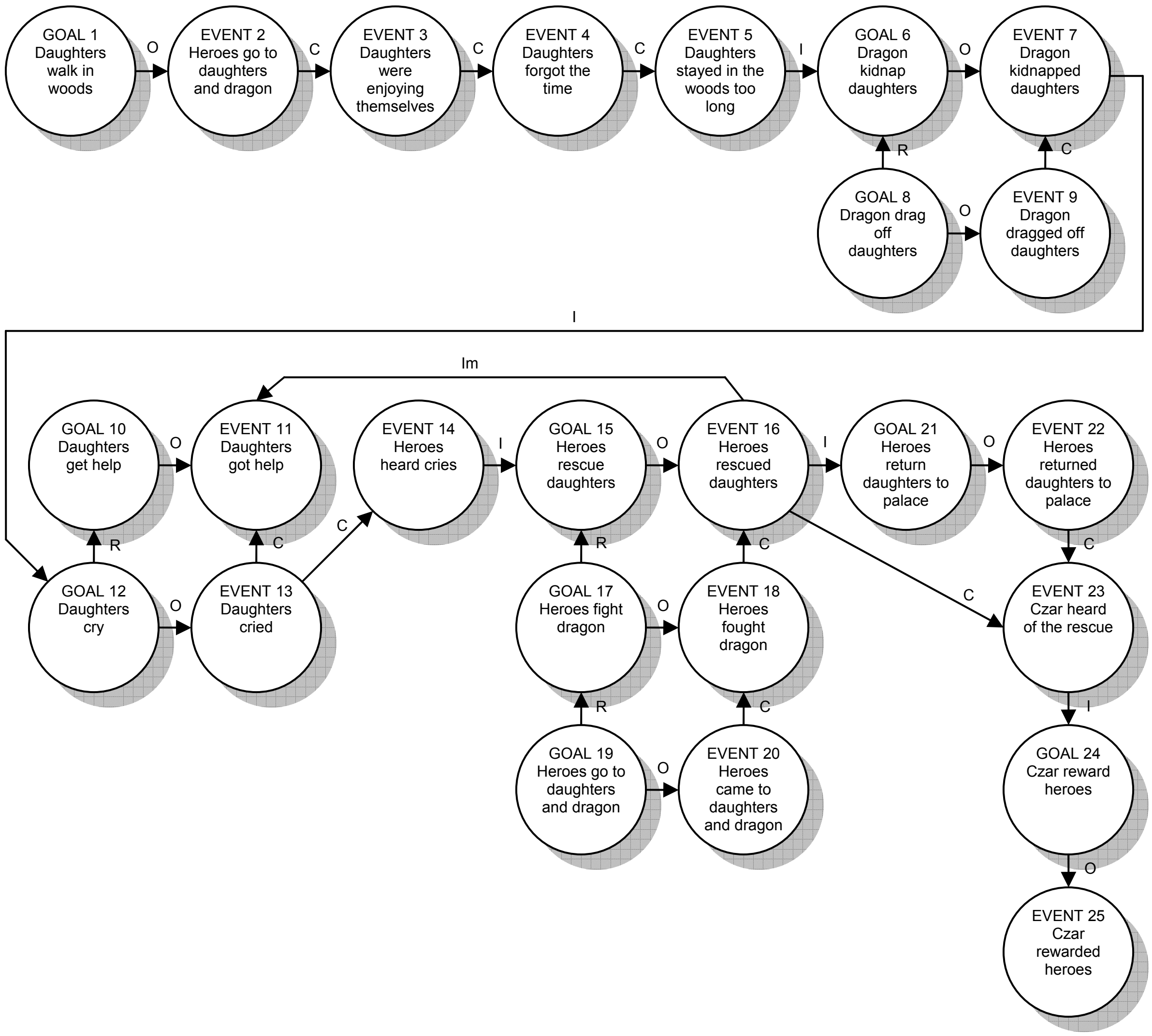

Figure 9.4. An example of a QUEST model of a story.

C. Because the daughters were walking in the woods (node 2).

D. Because the heroes fought the dragon (node 18).

The QUEST model defines arc search procedures for each type of question (e.g. why, how, when, enablement, and consequence). The arc search procedures, starting at the queried node, distinguish between legal answer nodes and illegal answer nodes. That is, only nodes reachable by the arc search procedures are legal answer nodes. The question above is a why question in which the event is not intentional. The search procedure is to follow backward C links. Therefore, answers (A) and (C) are legal answers. Of those two answers, (A) is preferred by the QUEST model because node 4 has a smaller structural distance from the queried node.

A QUEST knowledge structure – a representation of the cognitive structures held in the mind of a reader of a story – is a directed acyclical graph of events and goals. As such, QUEST knowledge structures are similar to fabula plans, which are also directed acyclical graphs of events and goals. Fabula plans representing the actions of a single story world character achieving a story goal can be directly transformed into a QUEST knowledge structure (Christian & Young 2004). The representation of a POCL plan[10] is relatively rigid compared to the expressivity of a QUEST knowledge structure (QKS). However, Christian and Young define a procedure by which a simple yet functional QKS can be derived from a POCL plan. The algorithm for transforming a POCL plan into a QKS is shown in Figure 9.5[11]. Christian and Young demonstrate that the QKS generated from a POCL plan significantly predict the goodness-of-answer judgments for "why" and "how" questions when arc search procedure was considered without structural distance.

Of particular interest to my evaluation of character believability in narratives generated by Fabulist is the *why* type of question in which an action is intentional. Referring back to the story in Figure 9.3, an example of this type of question is "Why did the heroes fight the dragon?" Intentional actions are represented in the QUEST model by a goal node that states what a character intends to accomplish and an event node that declares the character's attempt. The question queries the amalgamation of goal node 17 and event node 18 in the QKS for the story in Figure 9.4. The arc search procedure for why questions about intentional actions includes super-ordinate goals found by following forward reason arcs, backward initiate arcs, and backward outcome arcs (Graesser, Lang, & Roberts, 1991). Thus the only legal answer to the question is "Because the heroes wanted to rescue the daughters" (node 15).

---

[10] Christian and Young (2003) compare DPOCL plans to QUEST knowledge structures. DPOCL is a decompositional, partial order causal link planning algorithm (Young, Pollack, & Moore, 1994) that extends the conventional POCL algorithm by explicitly representing hierarchical relationships between abstract and primitive planning operators.

[11] The algorithm has been simplified from that originally given by Christian and Young (2003) to eliminate state nodes, which are unnecessary for GOA judgments for "why" questions.

Let n be the number of top-level goals of plan P and let m be the number of steps in P.

1. Create a total ordering o for the m steps in P that is consistent with the ordering constraints of P.
2. For each goal g_i of plan P for $i = 1, \ldots, n$, convert g_i into a goal node G_i.
3. For each plan step s_j for $j = 1, \ldots, m$ starting with the last step in o.
   a. Convert s_j into a goal node G_j and an event node E_j.
   b. Link G_j to E_j with an outcome arc.
4. For each causal link in P connecting two steps s_1 and s_2 with condition c, connect the event node E_1 to the event node E_2 with a consequence arc.
5. For each causal link $\prec s_1, p, q, s_2 \succ$ in P connecting two steps s_1 and s_2, connect G_1 to G_2 with a reason arc.

Figure 9.5. Algorithm for converting a POCL plan into a QUEST knowledge structure.

Given that the QUEST model is a significant predictor of goodness of answer for questions about stories, a QKS of a story can be used to evaluate a reader's comprehension character intentionality in the story. If a reader comprehends the intentionality of character actions in the story, the reader's GOA judgments for answers to questions about character intentions will be similar to the GOA judgments predicted by the QUEST model. Since a fabula plan can be transformed into a QKS that significantly predicts GOA judgments for "why" questions, reader comprehension of character intentionality in a story generated by Fabulist can be evaluated using the QKS generated by Christian and Young's algorithm for the fabula plan of the Fabulist-generated story.

The primary criticism of POCL planning for fabula generation is that they only consider the author's intentions – the outcome of the story – and neglect to consider the intentions of story world characters. Thus the characters in a story generated by a POCL planner will perform the actions necessary and sufficient for the outcome of the story to be achieved, regardless of whether the characters appear to have motivation to perform those actions. The significant contribution of the IPOCL algorithm is that it reasons about character intentions to determine whether character actions are believable. Subjects in the evaluation study are split into two conditions. The first condition – the *test condition* – evaluates subject perception of character intentionality in reading a story generated by the version of Fabulist that uses the IPOCL story planner module. The second condition – the *control condition* – evaluates subject perception of character intentionality in a story generated by the version of Fabulist that uses a conventional POCL story planner module. I test the perception of character believability in narratives generated by Fabulist using the IPOCL fabula planner component against the perception of character believability in narratives generated by Fabulist using a standard POCL fabula planner component. I compared the ability of subjects that read stories generated by both the IPOCL and POCL fabula planners to determine character intentions. The hypotheses of the experiment were as follows:

> **Hypothesis 1.** *If the QKS generated by Christian and Young's algorithm is a predictor of a reader's GOA judgements, then subjects in both conditions will have higher mean GOA judgments for question-answer pairs that are identified by QUEST as being "good" than question-answer pairs identified by QUEST as being "poor".*
>
> **Hypothesis 2.** *If a story generated by Fabulist is perceived to have character believability, then subjects in the test condition will have higher mean GOA judgment ratings for question-answer pairs identified by QUEST as being "good" than subjects in the control condition.*
>
> ***Hypothesis 3.*** *If a story generated by Fabulist is perceived to have character believability, then subjects in the test condition will have lower mean GOA judgment ratings for question-answer pairs identified by QUEST as being "poor" than subjects in the control condition.*

The first hypothesis, H1, tests to make sure that a QKS generated from a fabula plan is a significant predictor of a reader's GOA judgments. This is a verification of the results demonstrated by Graesser, Lang, and Roberts (1991) and Christian and Young (2004). The second and third hypotheses, H2 and H3, test that subjects in the test condition will be more aware of the intentions of the characters in the story and therefore be more accurately modeled by QUEST. Hypotheses H2 and H3 indirectly measure the character believability (as far as character intentionality is concerned) because the evaluation is a comparison between reader performance on question answering problems involving character intentionality and the established QUEST model.

9.2.1. Method

To determine whether subjects perceived character believability in stories generated by Fabulist, I used two versions of Fabulist to generate two similar narratives. Subjects were separated into groups and asked to read one of the stories rate the goodness of answer of question/answer pairs relating to the story they read. One version of Fabulist had a story planner component implementing the IPOCL algorithm, while the other used a conventional POCL planner. Both versions of Fabulist had identical discourse planner components based on the Longbow planner (Young, Moore, & Pollack, 1994), and identical template-based text realizer components. Both versions of Fabulist were initialized with identical parameters, including a description of a story world, a goal state to be achieved in the story world, and an action library that had a sufficient number of actions to successfully generate a story plan. The initialization parameters are described in Appendix B.

There is a woman named Jasmine. There is a king named Mamoud. This is a story about how King Mamoud becomes married to Jasmine. There is a magic genie. This is also a story about how the genie dies.

There is a magic lamp. There is a dragon. The dragon has the magic lamp. The genie is confined within the magic lamp.

King Mamoud is not married. Jasmine is very beautiful. King Mamoud sees Jasmine and instantly falls in love with her. King Mamoud wants to marry Jasmine. There is a brave knight named Aladdin. Aladdin is loyal to the death to King Mamoud. King Mamoud orders Aladdin to get the magic lamp for him. Aladdin wants King Mamoud to have the magic lamp. Aladdin travels from the castle to the mountains. Aladdin slays the dragon. The dragon is dead. Aladdin takes the magic lamp from the dead body of the dragon. Aladdin travels from the mountains to the castle. Aladdin hands the magic lamp to King Mamoud. The genie is in the magic lamp. King Mamoud rubs the magic lamp and summons the genie out of it. The genie is not confined within the magic lamp. King Mamoud controls the genie with the magic lamp. King Mamoud uses the magic lamp to command the genie to make Jasmine love him. The genie wants Jasmine to be in love with King Mamoud. The genie casts a spell on Jasmine making her fall in love with King Mamoud. Jasmine is madly in love with King Mamoud. Jasmine wants to marry King Mamoud. The genie has a frightening appearance. The genie appears threatening to Aladdin. Aladdin wants the genie to die. Aladdin slays the genie. King Mamoud and Jasmine wed in an extravagant ceremony.

The genie is dead. King Mamoud and Jasmine are married. The end.

Figure 9.6. Narrative text used in the test condition of the character believability evaluation study.

Fabulist was used to generate two stories of significant length. The first story was generated by a version of Fabulist with a fully functioning story planner module that implements the IPOCL algorithm. The IPOCL algorithm is a standard POCL algorithm enhanced to reason about character intentionality. The second story was generated by a version of Fabulist with a story planner module that implements a standard POCL algorithm that does not explicitly reason about character intentionality. Both versions of Fabulist used the same discourse planner module and media realizer module. The discourse planner module encapsulated an instance of the Longbow planner (Young, Moore, & Pollack, 1994) with a simplified discourse operator library. The media realizer used a simple template-matching approach to render the discourse plan operations into natural language.

Both the test condition story and the control condition story are generated from the same set of inputs. The stories differ due to the fact that the test condition story planner reasons about intervals of character intentionality and introduces additional motivating actions into the story to provide explanation for why characters act. The story in the test condition has 13 events and is shown in Figure 9.6, while the story in the control condition has 10 events and is shown in Figure 9.7. Figures 9.8 and 9.9 show fabula plan representations of the test

There is a woman named Jasmine. There is a king named Mamoud. This is a story about how King Mamoud becomes married to Jasmine. There is a magic genie. This is also a story about how the genie dies.

There is a magic lamp. There is a dragon. The dragon has the magic lamp. The genie is confined within the magic lamp.

There is a brave knight named Aladdin. Aladdin travels from the castle to the mountains. Aladdin slays the dragon. The dragon is dead. Aladdin takes the magic lamp from the dead body of the dragon. Aladdin travels from the mountains to the castle. Aladdin hands the magic lamp to King Mamoud. The genie is in the magic lamp. King Mamoud rubs the magic lamp and summons the genie out of it. The genie is not confined within the magic lamp. The genie casts a spell on Jasmine making her fall in love with King Mamoud. Jasmine is madly in love with King Mamoud. Aladdin slays the genie. King Mamoud is not married. Jasmine is very beautiful. King Mamoud sees Jasmine and instantly falls in love with her. King Mamoud and Jasmine wed in an extravagant ceremony.

The genie is dead. King Mamoud and Jasmine are married. The end.

Figure 9.7. Narrative text used in the control condition of the character believability evaluation study.

condition story and control condition story, respectively. Note that there are significant similarities between the two fabulas, making a comparison study possible. The set of actions in the fabula plan for the test condition is a superset of the actions in the fabula plan for the control condition. The difference is that the fabula plan for the test condition has frames of commitment and three additional character actions that were added to the plan to satisfy open motivations. There is one distinct action ordering difference between the two fabula plans. The action, `Fall-In-Love`, is temporally constrained to occur first in the fabula plan for the test condition. In the fabula plan for the control condition, `Fall-In-Love` is temporally under-constrained with respect to most other actions and happens to fall late in the plan because the final orderer has a bias towards keeping related actions temporally close. The final order used by Fabulist in both conditions attempts to position an action as close to (but before) actions that it satisfies conditions for.

I claim that the IPOCL algorithm generates narratives with explicit structures that facilitate audience understanding of character intentions. To test this assertion, I show that subjects that read a story generated by IPOCL perform significantly better in determining character intentions than subjects that read a story generated by a conventional POCL planner.

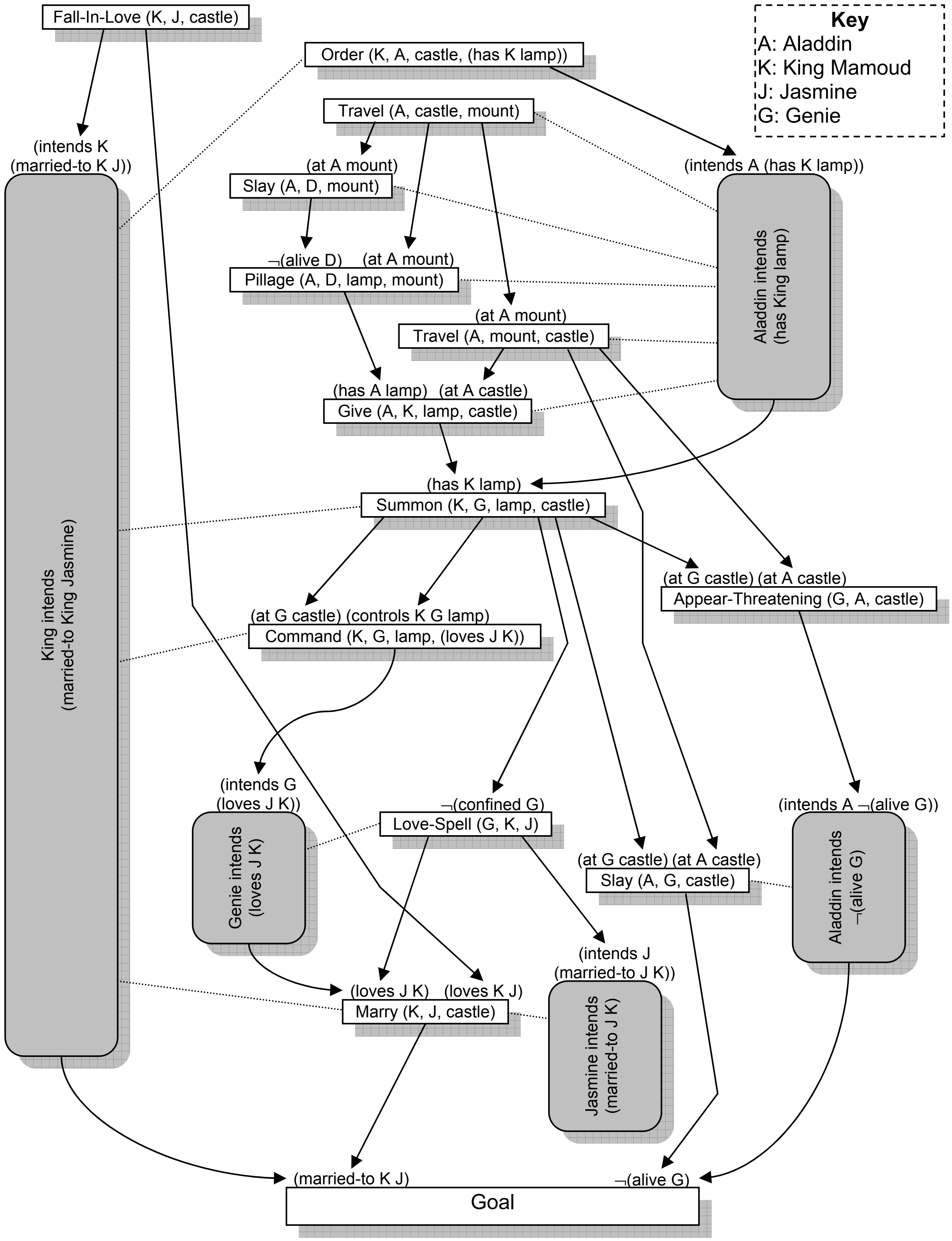

Figure 9.8. Fabula plan representation of the story used in the test condition of the character believability evaluation study.

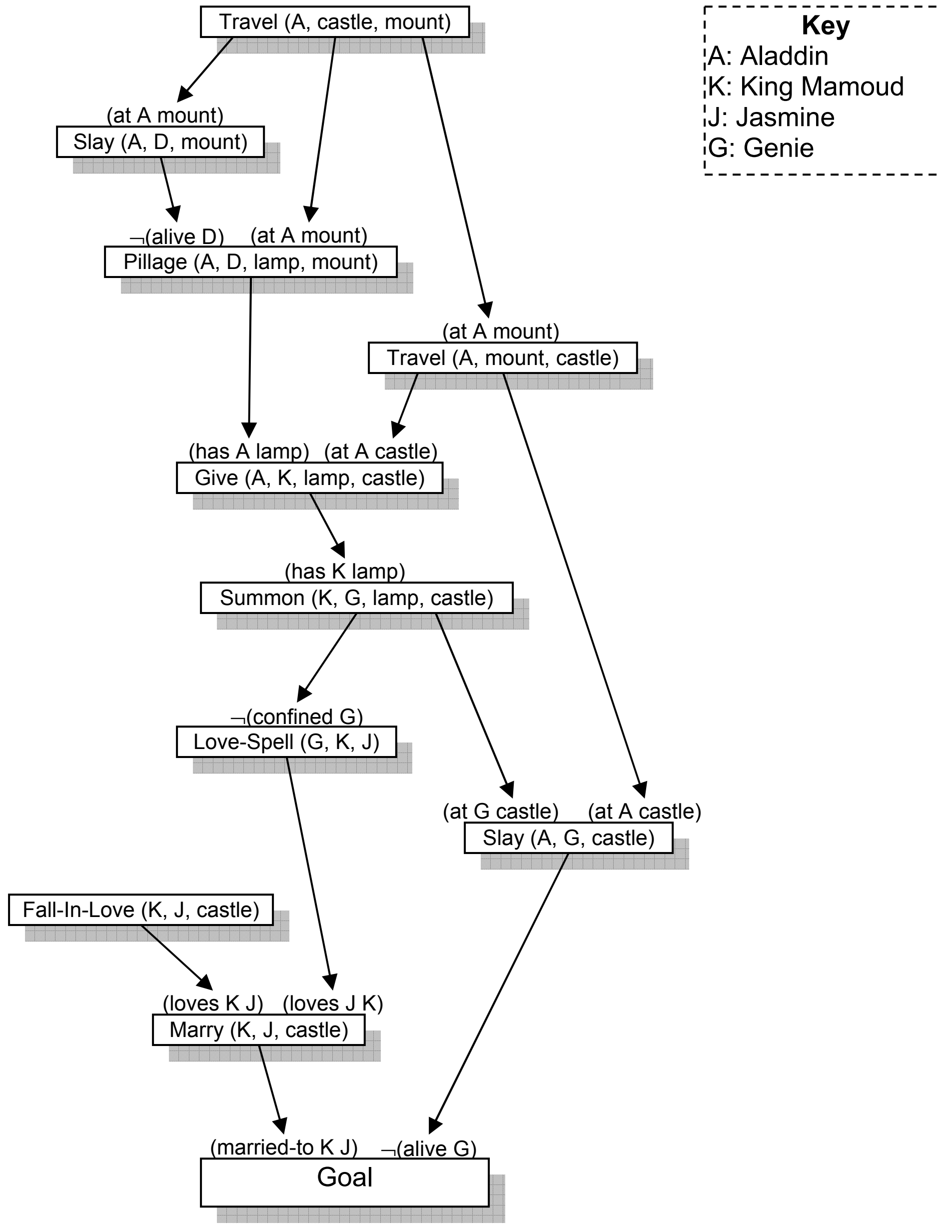

Figure 9.9. Fabula plan representation of the story used in the control condition of the character believability evaluation study.

9.2.1.1. Mapping IPOCL Plans to QUEST Knowledge Structures

Christian and Young's algorithm for generating a QKS from a plan (shown in Figure 9.5) has only been tested for POCL plans involving a single character. IPOCL plans, however, contain additional structures such as frames of commitment and motivation links that are not part of conventional plan representations. Consequently, the algorithm for generating a QKS from a plan must be extended to take into consideration IPOCL plans. I have added the following rules to the algorithm. Let *P* refer to the plan being transformed into a QKS.

- For each frame of commitment $c \in P$ such that *c* is motivated by step s_m and *c* is in service of step s_s and s_m and s_s are both members of the same interval of intentionality (thus character(s_m) = character(s_s)), then let G_m and E_m be the pair of nodes that represents s_m and G_s and E_s be the pair of node that represents s_s in the QKS. Connect G_m to G_s with a reason arc and connect E_m to E_s with a consequence arc.
- For each frame of commitment $c \in P$ such that *c* is motivated by step s_m and step s_f be the final step in *c*, let E_m be event node that represents s_m and G_f be the goal node that represents s_f in the QKS. Connect E_m to G_f with an initiate arc.

The first new rule handles the case where one character "contracts out" a task to another character in the sense of distributing a goal in the SharedPlans (Grosz & Kraus, 1996) formulation. A character a_1 is committed to a goal g_1 and performs an action, s_m, which motivates another character a_2 to commit to some subordinate goal g_2. The achievement of g_2 makes it possible for a_1 to perform additional actions in the world (of which one is step s_s) to achieve the super-ordinate goal, g_1. This rule ensures a coherent local goal hierarchy for a character that "contracts out" a task in the QKS. A local goal hierarchy is a sub-graph of a QKS that represents a plan that a character has to achieve some goal (van den Broek, 1988; Graesser, Lang, & Roberts, 1991). An example of a local goal hierarchy can be seen in nodes 15 through 20 in Figure 9.4. The second new rule converts motivation links in the IPOCL plan representation into initiate arcs in the QKS. Motivating steps cause characters to commit to a goal. That is, some event in the story world initiates a goal in a character.

In addition to the two new rules for converting a plan into a QKS, step 5 of the original algorithm (Figure 9.5) must be updated to reflect the fact that more than one character can take part in the fabula plan. The new step 5 of the algorithm reads as follows.

- For each causal link $\prec s_1, p, q, s_2 \succ \in P$ such that step s_1 and step s_2 are members of the same interval of intentionality (thus character(s_1) = character (s_2)) in *P*, let

G_1 be the goal node representing s_1 and G_2 be the goal node representing s_2 in the QKS. Connect node G_1 to node G_2 with a reason arc.

The new formulation of the step merely ensures that two characters' goal hierarchies are not conjoined when one character's actions cause another character to act. The arc search procedure remains unchanged: legal answer nodes are those found by following forward reason arcs, backward initiate arcs, and backward outcome arcs (Graesser, Lang, & Roberts, 1991).

Figures 9.10 and 9.11 show the QKS structures generated from the fabula plan in the test condition and fabula plan in the control condition, respectively. For clarity, the layout of nodes in both figures is biased to show local goal hierarchies of the characters instead of temporal progression, although the nodes in both figures have been numbered to reflect the order in which events in the story occur. Both QUEST knowledge structures were created using the modified algorithm described in this section, although the QKS for the fabula plan in the control condition did not use the two new rules since the fabula plan is a POCL plan. The modified algorithm has not been formally evaluated. However, if hypothesis H1 is supported by the evaluation results, then the QUEST knowledge structures will be shown to be significant predictors of GOA judgments for "why" questions. It is interesting to note that there are only five additional nodes in the QKS for the test condition than in the QKS for the control condition. The additional nodes result from the three steps in the fabula plan for the test condition that are not present in the fabula plan for the control condition. The most significant difference between the two QUEST knowledge structures is the relative complexity of the QKS for the test condition that arises from the inclusion of initiate arcs.

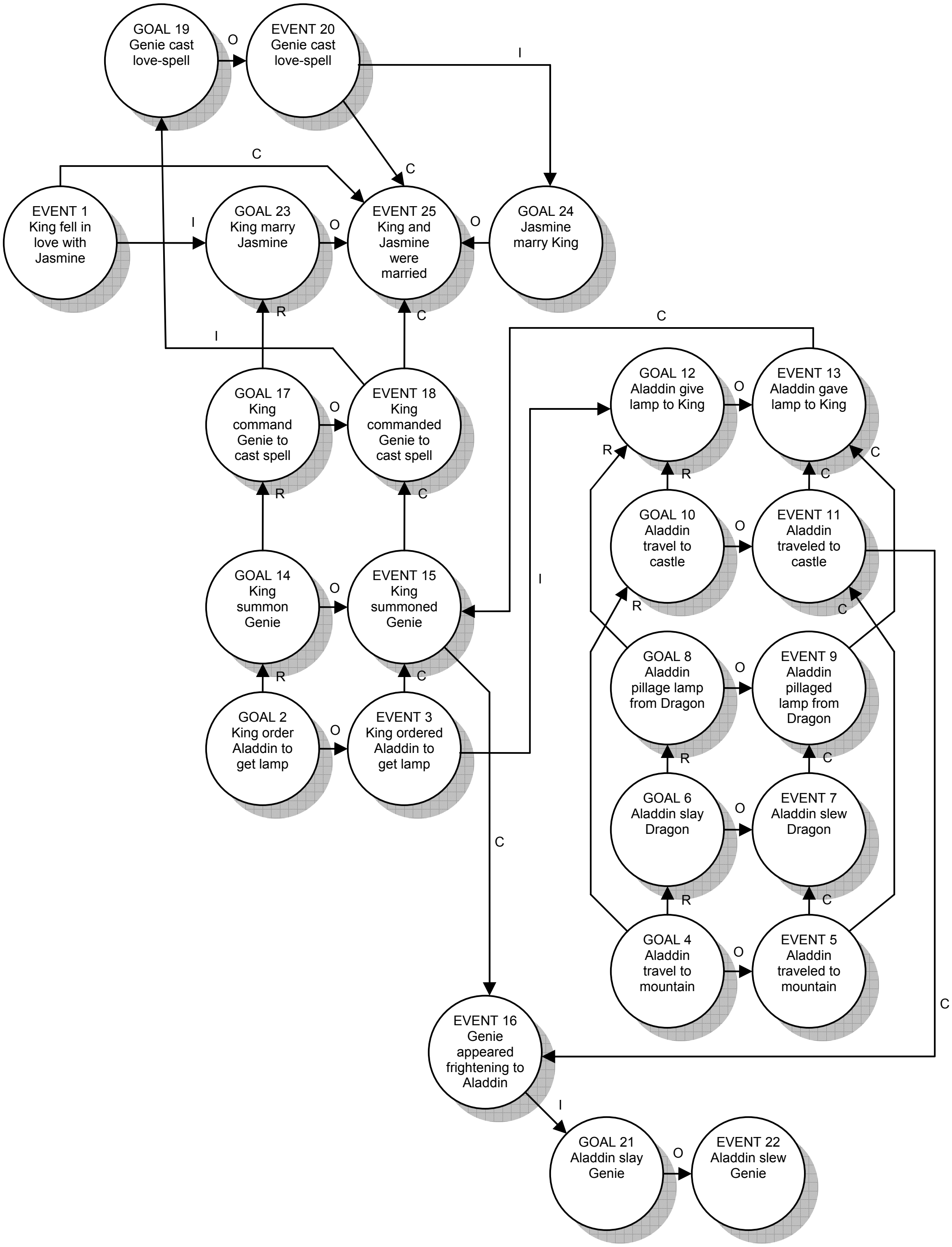

Figure 9.10. The QUEST knowledge structure for the IPOCL plan used in the test condition of the character believability evaluation study.

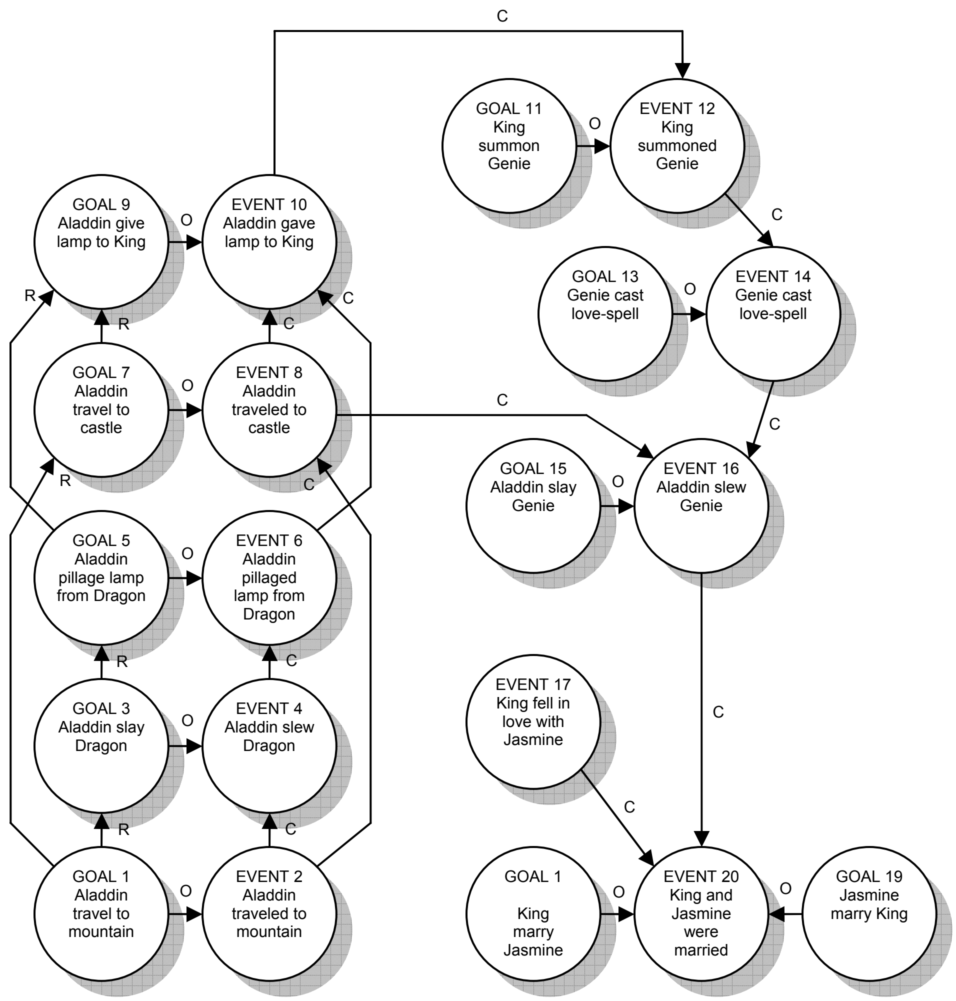

Figure 9.11. The QUEST knowledge structure for the IPOCL plan used in the control condition of the character believability evaluation study.

9.2.1.2. Testing Procedure

The evaluation was set up as a questionnaire in which subjects read a story and then made goodness-of-answer (GOA) judgments about pairs of question and answers. A question-answer pair has a "why" question about an intentional action performed by a character in the story. For example, the question, "Why did Aladdin slay the dragon?" might be paired with the answer, "Because King Mamoud ordered Aladdin to get the magic lamp for him." The subject was asked to rate the goodness of the answer for the given question on a four-point Likert scale ranging from "very bad answer" to "very good answer." The subjects were shown an example of a question-answer pair before the rating task began, but were not otherwise given a definition of “good” or “bad” or trained to make the judgment. Subjects rated the GOA of a question-answer pair for every combination of goal nodes in a QKS for the story. Subjects were asked to read the story text completely at least once before proceeding to the ratings task. Subjects were allowed to refer back to the original text at any time during the rating task. To shorten the length of the questionnaire, question-answer pairs were excluded in which the node from which the answer is based on temporally preceded the node from which the question is based on, except when an initiate arc might be involved. Each condition had a different number of question-answer pairs to rate due to the length of the respective fabula plans. The questionnaire for the test condition had 82 question-answer pairs while the questionnaire for the control condition only had 52. In addition to the importance ratings, I also asked subjects to estimate the number of times they had to refer back to the original text while performing the rating task.

Thirty-two undergraduate students in the Computer Science program at North Carolina State University participated in the study. All subjects were enrolled in the course, *Game Design and Development*, and were compensated for their time with five extra credit points on their final grade in the course. Half of the subjects were assigned randomly to each condition.

The QUEST model was used to predict the GOA judgment ratings that subjects would give each question-answer pair. Since Christian and Young (2004) found that the QKS generated from a plan was consistent with arc search procedure but not structural distance, I used the QKS for each condition to sort question-answer pairs in “good” and “poor” categories. If the subjects perceived character intentionality in the generated story, then the mean GOA judgment rating for each question-answer pair would be high for pairs that were in the “good” category and low for pairs that were in the “poor” category. A cluster analysis of the results from each condition showed that the GOA judgment ratings for one question-answer pair, corresponding to nodes 4 and 10 in the test condition and nodes 1 and 7 in the control condition, was anomalously low (1.3824 out of 4, when the QUEST model predicted a high GOA judgment rating). The question for this pair was “Why did Aladdin travel from the castle to the mountains?” and the answer was “Because Aladdin wanted to travel from the mountains to the castle.” The problem appears to come from the interleaving of local goal hierarchies for the Aladdin character (see Figure 9.10, nodes 4 through 13, and Figure 9.11, nodes 1 through 10). The consequence arcs and reason arcs are derived from the `(at …)` and `¬(at …)` conditions on the `Travel` operations in the fabula plans. The `at` and `¬at` conditions are necessary for the planner to reason about character location but apparently are not necessary for the QUEST model. Presumably, a reader has no problem recognizing that if a character travels from one location to another, then that character is at the latter location and not at the previous location without explicitly modeling this relationship.

9.2.2. Results

The first analysis compares the subjects' mean GOA judgments against the QUEST model predictions. The mean GOA rating for question-answer pairs in the "good" category was compared to the mean GOA rating for question-answer pairs in the "poor" category for each condition in the study. A standard one-tail t-test was used to compare the mean GOA rating of "good" question-answer pairs to the mean GOA rating of "poor" question-answer pairs in the test condition. The result of the t-test with 15 degrees of freedom for mean GOA ratings within the test condition yields $t = 35.8882$ ($p < 0.0005$). A standard one-tail t-test was used to compare the mean GOA rating of "good" question-answer pairs to the mean GOA rating of "poor" question-answer pairs in the control condition. The result of the t-test with 15 degrees of freedom for mean GOA ratings within the control condition yields $t = 13.7517$ ($p < 0.0005$). The results are shown in Table 9.2. These results indicate that the QUEST knowledge structures generated from the fabula plans in both conditions are significant predictors of GOA judgments for "why" questions when only arc search procedure is considered. The results of the analysis support hypothesis H1. It is interesting to note that, while both QUEST knowledge structures are strong predictors of GOA judgments ratings, the QKS in the test condition is a stronger predictor than the QKS in the control condition. Since hypothesis H1 is supported, I can move forward and compare the results of the test

Table 9.2. Results of the character believability evaluation study.

	Mean GOA rating for "good" question-answer pairs (standard deviation)	Mean GOA rating for "poor" question-answer pairs (standard deviation)
Test condition	3.1976 (0.1741)	1.1898 (0.1406)
Control condition	2.9912 (0.4587)	1.2969 (0.1802)

condition against the control condition.

The purpose of the evaluation study is to show that enhancements to the POCL algorithm for fabula planning significantly improve a reader's perception of character intentionality. To do that, hypotheses H2 and H3 need to be supported. The predictive ability of QUEST knowledge structures built from both POCL and IPOCL plan is confirmed. The support of hypotheses H2 and H3 demonstrates that readers have better comprehension of character intentions because they are in better agreement with the QUEST model. A standard one-tailed t-test was used to compare the mean GOA rating of "good" question-answer pairs in the test condition to the mean GOA rating of "good" question-answer pairs in the control condition. The result of the t-test with 15 degrees of freedom yields $t = 1.6827$ ($p < 0.0585$). The standard threshold for significance is $p < 0.05$. However, the threshold is not an absolute and a result of $p < 0.0585$ is close enough to be considered statistically significant. There are many ways to derive the degrees of freedom in a t-test. The 15 degrees of freedom used in this analysis is the most restrictive, meaning that if more degrees of freedom are considered, then the results become more significant. Hypothesis H2 is therefore supported. A standard one-tailed t-test was used to compare the mean GOA rating of "poor" question-answer pairs in the test condition to the mean GOA rating of "poor" question-answer pairs in the control condition. The result of the t-test with 15 degrees of freedom yields $t = 1.8743$ ($p < 0.05$). Hypothesis H3 is therefore supported.

9.2.3. Discussion

The fact that hypothesis H1 is supported means that the QKS built from a fabula plan generated by the story planner module using the IPOCL planning algorithm can be used by the QUEST model for predicting a reader's question-answering ability. Therefore, it is desirable that the subjects agree with the QUEST model. The fact that hypotheses H2 and H3 are supported indicates that when subjects in the test condition read an answer to a question about a story character's intentions, they are more likely to agree with the QUEST model on whether that answer is correct or not. Since subjects in the test condition are more in agreement with the QUEST model than subjects in the control condition, one can conclude that stories generated by a story planner implementing the IPOCL planning algorithm support a reader's comprehension of character intentionality better than stories generated by a story planner implementing a conventional POCL planner. Since perception of character intentionality affects the audience's perception of character believability, according to the model of character believability in Section 3.2.2, the IPOCL fabula planning algorithm is a significant improvement over conventional planning for fabula generation in terms of providing character believability.

The analysis presented above determines a statistical correlation that proves the hypotheses. However, the analysis does not clearly identify the reasons for the correlation. What does a fabula plan generated by IPOCL have that supports the audience's perception of character intentionality that a fabula plan generated by a conventional POCL plan does not? The analysis tells us that a reader of a story generated by a story planner using IPOCL will have a more accurate understanding of character intentionality, but the analysis does not tell us why. I break the results of the evaluation study down a different way. There are many question-answer pairs that are common between the test condition questionnaire and the control condition questionnaire. The similarities are due to the fact that the fabula plan in the control condition contains a subset of the actions that the fabula plan in the control condition contains. If better comprehension of character intentionality is a result of the way an IPOCL fabula planner adds actions to the fabula to motivate why a character is committed to a particular goal, then GOA judgments between question-answer pairs that are common between conditions will not be significantly different. Table 9.3 shows the results when I only consider the question-answer pairs shared between the two conditions. It is visibly obvious that the results are much closer. In fact, a t-test to compare the mean GOA rating for "good" question-answer pairs in the test condition to the mean GOA rating for "good" question-answer pairs in the control condition indicates that there is no significant difference. That is, had subjects in the test condition only considered the same actions that the subjects

Table 9.3. Results of the character believability study when only shared question-answer pairs are considered.

	Mean GOA rating for "good" question-answer pairs (standard deviation)	Mean GOA rating for "poor" question-answer pairs (standard deviation)
Test condition	2.9500 (0.2046)	1.2017 (0.1548)
Control condition	2.9912 (0.4587)	1.2969 (0.1802)

in the control condition were exposed to, there would be no difference between groups of subjects. A t-test to compare the mean GOA rating for "poor" question-answer pairs in the test condition to the mean GOA rating for "poor" question-answer pairs in the control condition has a result of $t = 1.6030$ ($p < 0.1$). This means that there is some evidence that the GOA ratings for shared question-answer pairs that are considered "poor" is lower for the test condition than the control condition, but not at the traditionally accepted $p < 0.05$ threshold.

One conclusion that can be drawn from the additional analysis is that the additional story elements that were created by the IPOCL fabula planner enable the reader to more readily reject hypotheses about character intentions. I assume that the reader is actively engaged in the story and is attempting to use inferential reasoning to determine the intentions of characters in the story world. If the intentions of a character are ambiguous to the reader, the reader cannot reject any hypothesis that explains the motivation of that character. When asked to rate the GOA of a "why" question-answer pair in which the answer is a poor explanation of the character's intention, the reader cannot definitively reject the explanation. Consequently, the mean GOA of "poor" question-answer pairs for control subjects tend towards the middle of the scale. Even though subjects in the test condition are rating the same question-answer pairs, the additional information they have been given helps them be more decisive when rejecting poor explanations of character intentions. The results suggest that there is ambiguity about character intentions in the story in the control condition and this ambiguity causes readers to postulate explanations to explain character intentionality or at least not to discard any plausible explanation of character intentionality. Storytelling is fundamentally a form of communication. Regardless of the content of any communicative act, one will always be able to provide a more or less plausible explanation of the meaning (Sadock, 1990). This is true even when the content of a sequence of communicative acts is the fabula of a story and the meaning is the intentionality of a story character.

Surprisingly, the same conclusion can not be made about GOA ratings for "good" question-answer pairs. One possible explanation for this is that it is easier in general for a reader to identify good explanations of character intentionality than to reject poor explanations of character intentionality. The standard deviation of answers in the control condition, however, is much larger than the standard deviation in the test condition indicating more variability in how subjects in the control condition rated the GOA of good question-answer pairs.

9.3. Summary

This chapter describes two evaluation studies on stories generated by Fabulist. The first study determines whether readers perceive a story generated by Fabulist to be plot coherent. The study asks subjects to read a story generated by Fabulist that has additional, non-coherent aspects added. Since the subjects ranked sentences in the story as highly important when the events they describe had meaning and relevance to the outcome of the story and ranked sentences in the story as highly unimportant when the events the describe had no relevance to the outcome of the story, I am able to conclude that the story is plot coherent. The second study determines whether readers perceive a story generated by Fabulist to have character believability. The study asked subjects to read one of two stories generated by Fabulist. The first story was generated by a version of Fabulist that used the IPOCL planning algorithm to generate the fabula. The second story was generated by a version of Fabulist that used a standard POCL planning algorithm to generate the fabula. Subjects made determinations about character intentions that more closely resembled the predicted answers when they read the first story. This enables me to conclude that the first story has more character believability than the second character. Consequently, Fabulist with IPOCL is better at generating stories than Fabulist with a standard planner.

Chapter 10

Conclusions

In this chapter, I conclude with the major contributions and findings of the work presented in this dissertation. The objective of the research presented here is to use a planning approach to generating narratives that have strong plot coherence and strong character believability. A review of the literature suggests that story generation systems can be roughly split into two categories: character-centric systems and author-centric systems. Both categories have their strengths and weaknesses when it comes to generating stories with plot coherence and character believability. Specifically, character-centric systems, which focus on the autonomy of characters in a story-world simulation, more reliably generate stories with strong character believability and less reliably generate stories with plot coherence. On the other hand, author-centric systems, which model the processes by which an author reasons about the structure of stories, more reliably generate stories with strong plot coherence and less reliably generate stories with character believability. Planning is an author-centric approach to story generation that treats the construction of a story as problem solving. The story is a sequence of events that transforms the story world from an initial configuration into a final outcome.

As an author-centric technique, planning is not well suited to generating stories with character believability. The contribution of my research is to show how planning makes a good model of dramatic authoring and to provide ways in which the standard planning algorithm can be extended so that a story generation system based on planning can generate stories with both strong plot coherence and strong character believability.

10.1. Planning as a Model of Dramatic Authoring

Partially-ordered planning is an effective model of dramatic authoring for two reasons. The first reason is that partially-ordered plan structures including actions to be performed by multiple characters have many similarities to fabulas. The fabula of a narrative is an enumeration of all the events that occur in the story world from the time the story begins to the time the story ends. The temporal ordering of the events in the fabula corresponds to the order the actions actually occur in, as opposed to the order they are told in. Furthermore, the fabula is relatively complete, meaning that it contains events that must occur for causal continuance but are not necessarily told to the audience. Finally, while the discourse process of telling a story is typically linear in nature, events in the fabula can occur concurrently, making the partial ordering of plan structures an important feature. The effectiveness of the plan structure as a model of fabula is evident in the similarities of partially-ordered plans to the cognitive structures held in the minds of those who were told a story (Christian & Young, 2004). People who are told stories cognitively model the content of a narrative and not the discourse through which the narrative was related (Thorndyke, 1977).

The second reason that partially-ordered planning makes an effective model of dramatic authoring is that there is a strong correlation between search-based planning algorithms and the process by which human authors write stories. Planners are general problem-solvers. In the case of story generation, a story planner solves the problem of constructing a sequence of events that satisfies the user-defined outcome of the story world. Egri (1960) describes the process of authoring dramatic works as the search for a set of characters that can prove a given premise if allowed to act freely in the story world. If the characters do not act in a way that satisfies the premise or must be forced to act contrary to their natures to satisfy the premise, the author should backtrack and begin again with different characters. If a planner can construct a story plan in which the characters act believably (e.g. perform the actions they would have chosen to perform had they been acting autonomously), then that planner can be used to determine whether a set of characters can act to satisfy the premise of they story. However, the planner must also search through the set of possible story worlds for an appropriate set of characters.

The model of dramatic authoring described by Egri (1960) states that the author should backtrack and revise characters when it becomes apparent that a story is not going to prove the given premise. Egri assumed that backtracking meant returning to the beginning of the story and rewriting. However, a least-commitment planner can backtrack to an intermediate stage of authoring (constructing a plan) because the planner does not commit to anything until it is absolutely necessary. Fully determine characters constrain the stories that a story planner can generate. If characters are fully specified, the planner must work with the characters it is given and must either work around the constraints, completely disregard stories in which characters act contrary to their declared nature, or construct a story in which characters act "out of character." However, if a character's traits are left undetermined, then a planner can search for a possible world (or set of possible worlds) in which characters are defined in such a way that the plot unfolds naturally. The Initial State Revision planning algorithm as story author determines characters' traits in the initial world state in a least-commitment approach. ISR meets the requirements of a computational model of dramatic authoring because it can backtrack to find character descriptions that are appropriate. By extending the POCL algorithm to handle an initial state that has undetermined sentences describing the world, planning indeed becomes a process that is equivalent to one way in which human authors create stories.

The model of dramatic authoring does not describe how to ensure that a story has strong plot coherence or strong character believability. However, the very nature of the POCL planning algorithm, particularly the way in which it uses causal necessity to select action schemata, means that a limited form of plot coherence – story coherence – is achieved as a side-effect of the plan construction process. The actions in a partially-ordered plan are related to each other through causal links which specify that earlier actions have effects which establish the conditions necessary for later actions to execute successfully. The nature of the causal relationships ensures that all actions (not just the main actions) are part of a causal chain that terminates in the outcome of the story; the relationship of each action is literally represented in the structure of the plan. The plan structure and planning algorithm do not need to be extended to be able to construct plans that have story coherence. An empirical evaluation confirms that readers of stories generated by Fabulist form cognitive representations of the story that indicate perception of story coherence.

10.2. Planning for Character Believability

Character believability, however is more problematic since the concept of character believability does not directly correlate to any structural elements in the plan or any process in the planning algorithm. Part of the reason for this is that there is no concise definition of character believability. Character believability refers to the numerous elements that allow a character to achieve the "illusion of life" (Bates, 1994). The research presented in this dissertation focuses on two of those elements, intentionality and consistency of character behavior, although there are many other elements such as character appearance.

A conventional planner, such as a POCL planner, inserts operations into the plan that are sufficient and necessary to achieve some goal. Systems that use planners assume that the goal state is intended by the agents that execute the plan. Therefore, a POCL planner could only construct story plans in which all characters intend the outcome and cooperate to perform actions that achieve the goal. The type of stories is very limited by the assumption of intentionality concerning planning. The IPOCL planning algorithm, on the other hand, does not rely on the assumption of intentionality and is therefore able to construct story plans in which characters do not necessarily intend the outcome and do not necessarily cooperate on any goal. Story world characters do not necessarily start out with any intentions. Instead, events occur in the story plan that motivates the character to commit to individual goals.

In order to expand the expressivity of stories that can be told, Fabulist expands the representation of the partially-ordered plan to include information about character's individual goal commitments. The IPOCL planning algorithm is correspondingly expanded to be able to reason about goals characters might have and to find reasons for why those characters are motivated to commit to those goals. IPOCL thus searches a larger search space than a conventional POCL planner. The larger search space includes plans in which characters actions can be perceived to be intentional even though the characters do not necessarily intend the outcome of the story. The perception of character intentionality is an important part of establishing character believability in a story. An empirical evaluation confirms that stories generated by Fabulist using the IPOCL algorithm to generate the fabula support the reader's comprehension of character intentionality. However, planning for stories with character believability increases the branching factor and depth of the space of plans that must be searched by the planner.

Consistency of behavior is also important for character believability. Without consistency, characters are prone to behave in a schizophrenic manner, performing behavior after behavior without portraying any coherent sense of identity (Sengers, 2000b). Consequently, agent consistency is difficult to achieve in a system that pieces together sequences of atom actions. One technique for achieving consistency in behavior is to evaluate character actions against some definition of personality. If an action schema were to encode the preferred personality traits of a character that would perform that action in its preconditions (or effects as in (Rizzo, 1999)), then characters that did not have the required traits would be excluded from performing that action. Characters would be effectively barred from action "out of character" because they could only perform actions whose specifications matched the character's set of trait.

I extend the action schema representation to use recommendation conditions that specify preference over characters that can perform an action. Recommendation conditions are similar to preconditions, except that recommendations do not need to be satisfied. For the planner to not satisfy a recommendation means that a character is acting out of character in that story plan. The POCL planning algorithm is extended such that whenever a recommendation flaw is selected for repair, the planner splits the search space into a sub-tree in which the recommendation is satisfied and a sub-tree in which the recommendation is not satisfied. Heuristics favor plans in which more recommendations are satisfied so that the story planner favors stories in which characters act consistently but is still able to consider story plans in which characters act "out of character" when there is no other way of completing a story plan. By expanding the plan representation and planning algorithm, Fabulist is capable of constructing a wider range of stories in which characters can act both consistently and inconsistently.

References

Allen, J.F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11), 832-843.

Allport, G.W. & Odbert, H.S. (1936). Trait names: A psycho-lexical study. *Psychological Monographs*, 47 (211), 1-171.

Aristotle. (1992). *The Poetics*. (T. Buckley, trans.). Buffalo, NY: Prometheus Books. (Original work published 350 B.C.E.)

Aylett, R. (1999). Narrative in virtual environments – Towards emergent narrative. In M. Mateas & P. Sengers (Eds.) *Narrative Intelligence: Papers from the 1999 Fall Symposium* (pp. 83-89). Menlo Park, CA: American Association for Artificial Intelligence.

Aylett, R. (2000). Emergent narrative, social immersion and "storification." *Proceedings of the 1st International Workshop on Narrative and Interactive Learning Environments*, 35-44.

Bailey, P. (1999). Searching for storiness: Story-generation from a reader's perspective. In M. Mateas & P. Sengers (Eds.) *Narrative Intelligence: Papers from the 1999 Fall Symposium* (pp. 157-163). Menlo Park, CA: American Association for Artificial Intelligence.

Bal, M. (1997). *Narratology: Introduction to the Theory of Narrative*. Toronto: University of Toronto Press.

Bates, J. (1992). Virtual reality, art, and entertainment. *Presence: The Journal of Teleoperators and Virtual Environments*, 1(1), 133-138.

Bates, J. (1994). The role of emotion in believable agents. *Communications of the ACM*, 37(7), 122-125.

Black, J.B. & Bower, G.H. (1980). Story understanding as problem-solving. *Poetics*, 9, 223-250.

Black, J.B. & Wilensky, R. (1979). An evaluation of story grammars. *Cognitive Science*, 3, 213-230.

Blair, D. & Meyer, T. (1997). Tools for an interactive virtual cinema. In R. Trappl & P. Petta (Eds.) *Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents* (pp. 83-91). New York: Springer.

Blumberg, B.M. & Galyean, T.A. (1995). Multi-level direction of autonomous creatures for real-time virtual environments. *Proceedings of the 22nd Annual Conference on Computer Graphics*, 39-46.

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

Bratman, M.E. (1987). *Intentions, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press.

Bratman, M.E. (1990). What is intention? In P.R. Cohen, J. Morgan, & M.E. Pollack (Eds.) *Intentions in Communication* (pp. 15-31). Cambridge, MA: The MIT Press.

Bratman, M.E., Israel, D.J., & Pollack, M.E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(4), 349-355.

Brown, A.L. & Smiley, S.S. (1977). Rating the importance of structural units of prose passages: A problem of metacognitive development. *Child Development*, 48, 1-8.

Bruckman, A. (1990). *The Combinatorics of Storytelling:* Mystery Train *Interactive*. Unpublished manuscript.

Bruner, J. (1990). *Acts of Meaning*. Cambridge, MA: Harvard University Press.

Bruner, J. (1991). The narrative construction of reality. *Critical Inquiry*, 18(1), 1-21.

Callaway, C.B. (2000). *Narrative prose generation*. Ph.D. Dissertation, Department of Computers Science, North Carolina State University.

Carberry, S. (2001). Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11 (1-2), 31-48.

Cavazza, M., Charles, F., & Mead, S.J. (2001). Characters in search of an author: AI-based virtual storytelling. *Proceedings of the 1st International Conference on Virtual Storytelling*, 145-154.

Chapman, D. (1987). Planning for conjunctive goals. *Artificial Intelligence*, 32(3), 333-377.

Charles, F., Lozano, M., Mead, S., Bisquerra, A., & Cavazza, M. (2003). Planning formalisms and authoring in interactive storytelling. *Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 216-225.

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

Christian, D.B. & Young, R.M. (2004). Comparing cognitive and computational models of narrative structure. *Proceedings of the 19th National Conference on Artificial Intelligence*, 385-390.

Cohen, P.R. & Levesque, H.J. (1990). Persistence, Intention, and Commitment. In P.R. Cohen, J. Morgan, & M.E. Pollack (Eds.) *Intentions in Communication* (pp. 33-69). Cambridge, MA: The MIT Press.

Costa, P.T. & McCrae, R.R. (1985). *The NEO Personality Inventory Manual*. Odessa, FL: Psychological Assessment Resources.

Dautenhahn, K. (2003). Stories of lemurs and robots: The social origin of story-telling. In M. Mateas & P. Sengers (Eds.) *Narrative Intelligence* (pp. 63-90). Philadelphia: John Benjamins Publishing.

Dehn, N. (1981). Story generation after Tale-Spin. *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 16-18.

Egri, L. (1960). *The Art of Dramatic Writing: Its Basis in the Creative Interpretation of Human Motives*. New York: Simon & Schuster.

Elhadad, M. (1993). *Using Argumentation to Control Lexical Choice: A Functional Unification Implementation*. Ph.D. Dissertation, Department of Computer Science, Columbia University.

Ford, M.E. (1992). *Motivating Humans: Goals, Emotions, and Personal Agency Beliefs*. Newbury Park, CA: Sage Publications.

Freytag, G. (1968). *Technique of the Drama: An Exposition of Dramatic Composition and Art* (E.J. MacEwan, trans.). New York: B. Blom. (Original work published 1863)

Galyean, T.A. (1995). *Narrative Guidance of Interactivity*. Ph.D. Dissertation, MIT Media Arts and Sciences Program.

Gerrig, R.J. (1993). *Experiencing Narrative Worlds: On the Psychological Activities of Reading*. New Haven: Yale University Press.

Gerrig, R.J. & Bernardo, D. (1994). Readers as roblem-solvers in the experience of suspense. *Poetics*, 22, 459-472.

Gordon, A., van Lent, M., van Velsen, M., Carpenter, P., & Jhala, A. (2004). Branching storylines in virtual reality environments for leadership development. *Proceedings of the 16th Innovative Applications of Artificial Intelligence Conference*, 844-851.

Graesser, A.C., Lang, K.L., & Roberts, R.M. (1991). Question answering in the context of stories. *Journal of Experimental Psychology: General*, 120(3), 254-277.

Grice, H.P. (1975). Logic and conversation. In P. Cole and J.L. Morgan (Eds.) *Syntax and Semantics III: Speech Acts* (pp. 41-58). New York: Academic Press.

Grosz, B.J. & Kraus, S. (1996). Collaborative plans for complex group action. *Artificial Intelligence*, 86(2), 269-357.

Grosz, B.J. & Sidner, C.L. (1990). Plans for discourse. In P.R. Cohen, J. Morgan, & M.E. Pollack (Eds.) *Intentions in Communication* (pp. 417-444). Cambridge, MA: The MIT Press.

Hayes-Roth, B. & van Gent, R. (1996). *Story-making with improvisational puppets* (Technical Report KSL-96-05). Palo Alto, CA: Knowledge Systems Laboratory, Stanford University.

Hayes-Roth, B., van Gent, R., & Huber, D. (1997). Acting in Character. In R. Trappl & P. Petta (Eds.), *Creating Personalities for Synthetic Characters: Towards Autonomous Personality Agents* (pp. 92-112). New York: Springer.

Herman, D. (2002). *Story Logic: Problems and Possibilities of Narrative*. Lincoln, NE: University of Nebraska Press.

Jhala, A.H. (2004). *An Intelligent Cinematic Camera Planning System for Dynamic Narratives*. Masters Thesis, Department of Computer Science, North Carolina State University.

Johnstone, K. (1981). *Impro: Improvisation for the Theatre*. London: Routledge.

Juul, J. (2001). Games telling stories – A brief note on games and narratives. *International Journal of Computer Game Research* [Online serial], 1(1). Available: http://www.gamestudies.org/0101/juul-gts/.

Kelso, M.T., Weyhrauch, P., & Bates, J. (1993). Dramatic presence. *Presence: The Journal of Teleoperators and Virtual Environments*, 2(1), 1-15.

Knoblock, C.A. (1994). Generating parallel execution plans with a partial-order planner. *Proceedings of the 2nd International Conference on Artificial Intelligence Planning Systems*, 98-103.

Lambert, L. & Carberry, S. (1991). A tripartite plan-based model of dialogue. *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, 47-54.

Lamstein, A. & Mateas, M. (2004). Search-based drama management. In D. Fu, S. Henke, & J. Orkin (Eds.) *Challenges in Game Artificial Intelligence: Paperes from the AAAI Workshop* (pp. 103-107). Menlo Park, CA: American Association for Artificial Intelligence.

Laurel, B. (1991). *Computers as Theatre*. Reading, MA: Addison-Wesley.

Lebowitz, M. (1984). Creating characters in a story-telling universe. *Poetics*, 13, 171-194.

Lebowitz, M. (1985). Story-telling as planning and learning. *Poetics*, 14, 483-502.

Lebowitz, M. (1987). Planning stories. *Proceedings of the 9th Annual Conference of the Cognitive Science Society*, 234-242.

Lehnert, W.G. (1982). Plot units: A narrative summarization strategy. In Lehnert, W.G. & M.H. Ringle (Eds.) *Strategies of Natural Language Processing* (pp. 375-412). Hillsdale, NJ: Lawrence Erlbaum Associates.

Lester, J., Voerman, J., Towns, S., & Callaway, C. (1999). Deictic believability: Coordinating gesture, locomotion, and speech in lifelike pedagogical agents. *Applied Artificial Intelligence*, 13(4-5), 383-414.

Loyall, A.B. (1997). *Believable Agents: Building Interactive Personalities*. Ph.D. Dissertation, Carnegie Mellon University.

Maes, P., Darrell, T., Blumberg, B., & Pentland, A. (1995). The ALIVE system: Full-body interaction with autonomous agents. *Proceedings of the 1995 Conference on Computer Animation*, 11-18.

Mandler, J. (1984). *Stories, Scripts, and Scenes: Aspects of Schema Theory*. Hillsdale, NJ: Erlbaum.

Mann, W.C. & Thompson, S.A. (1987). *Rhetorical Structure Theory: A Theory of Text Organization* (Technical Report ISI/RS-87-190). Marina del Rey, CA: Information Sciences Institute, University of Southern California.

Mao, W. & Gratch, J. (2004). Social judgment in Multiagent Interactions. *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems*, 210-217.

Marcella, S.C., Johnson, W.L., & LaBore, C. (2000). Interactive pedagogical drama. *Proceedings of the 4th International Conference on Autonomous Agents*, 301-308.

Mateas, M. (1997). *An Oz-centric review of interactive drama and believable agents* (Technical Report CMU-CS-97-156). Pittsburgh, PA: School of Computer Science, Carnegie Mellon University.

Mateas, M. (2002). *Interactive Drama, Art and Artificial Intelligence*. Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University.

Mateas, M. & Sengers, P. (1999). Narrative Intelligence. In M. Mateas & P. Sengers (Eds.) *Narrative Intelligence: Papers from the 1999 Fall Symposium* (pp. 1-10). Menlo Park, CA: American Association for Artificial Intelligence.

Mateas, M. & Stern, A. (2000). Towards integrating plot and character for interactive drama. In K. Dautenhahn (Ed.) *Socially Intelligent Agents: The Human in the Loop: Papers from the 2000 Fall Symposium* (pp. 113-118). Menlo Park, CA: American Association for Artificial Intelligence.

Mateas, M. & Stern, A. (2002). *Architecture, Authorial Idioms and Early Observations of the Interactive Drama Façade* (Technical Report CMU-CS-02-198). Pittsburgh, PA: School of Computer Science, Carnegie Mellon University.

McDermott, D. (1987). A critique of pure reason. *Computational Intelligence*, 3, 151-160.

McKee, R. (1997). *Story: Substance, Structure, Style, and the Principles of Screenwriting*. New York: ReganBooks.

Meehan, J.R. (1976). *The Metanovel: Writing Stories by Computer*. Ph.D. Dissertation, Yale University.

Meehan, J.R. (1977). Tale-Spin: An interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, 91-98.

Moore, J.D. & Paris, C. (1993). Planning text for advisory dialogues: Capturing intentional, rhetorical, and attentional information. *Computational Linguistics*, 19(4), 651-694.

Moore, J.D. & Pollack, M.E. (1992). A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics*, 18(4), 537-544.

Mott, B., Callaway, C., Zettlemoyer, L., Lee, S., & Lester, J. (1999). Towards narrative-centered learning environments. In M. Mateas & P. Sengers (Eds.) *Narrative Intelligence: Papers from the 1999 Fall Symposium* (pp. 78-82). Menlo Park, CA: American Association for Artificial Intelligence.

Norling, E. & Sonenberg, L. (2004). Creating interactive characters with BDI agents. *Proceedings of the 2004 Australian Workshop on Interactive Entertainment*, 69-76.

Osborn, B.A. (2002). *An Agent-Based Architecture for Generating Interactive Stories*. Doctoral Dissertation, Naval Postgraduate School.

Pednault, E.P.D. (1986). *Toward a Mathematical Theory of Plan Synthesis*. Ph.D. Dissertation, Stanford University.

Penberthy, J.S. & Weld, D. (1992). UCPOP: A sound, complete, partial-order planner for ADL. *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning*, 103-114.

Pérez y Pérez, R. & Sharples, M. (2001). Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence*, 13, 119-139.

Perlin, K. & Goldberg, A. (1996). Improv: A system for scripting interactive actors in virtual worlds. *Proceedings of the 23rd International Conference on Computer Graphics and Interactive Techniques*, 205-216.

Pervin, L.A. (1993). *Personality: Theory and Research* (6th ed.). New York: Wiley & Sons.

Pollack, M.E. (1992). The uses of plans. *Artificial Intelligence*, 57(1), 43-68.

Prince, G. (1987). *A Dictionary of Narratology*. Lincoln, NE: University of Nebraska Press.

Reilly, W.S.N. (1996). *Believable Social and Emotional Agents*. Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University.

Riedl, M.O. (2004). *Equivalence between narrative mediation and branching story graphs* (Technical Report TR04-004). Liquid Narrative Group, Department of Computer Science, North Carolina State University.

Riedl, M.O., Saretto, C.J., & Young, R.M. (2003). Managing interaction between users and agents in a multiagent storytelling environment. *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi Agent Systems*, 741-748.

Riedl, M.O. & Young, R.M. (2004). A planning approach to story generation for history education. *Proceedings of the 3rd International Conference for Narrative and Interactive Learning Environments*, 41-48.

Rizzo, P., Veloso, M.M., Miceli, M., & Cesta, A. (1999). Goal-based personalities and social behaviors in believable agents. *Applied Artificial Intelligence*, 13, 239-272.

Roussesau, D. & Hayes-Roth, B. (1998). A social-psychological model for synthetic actors. *Proceedings of the 2nd International Conference on Autonomous Agents*, 165-172.

Rumelhart, D.E. (1975). Notes on a schema for stories. In D.G. Bobrow & A. Collins (Eds.) *Representation and Understanding: Studies in Cognitive Science* (pp. 211-236). New York: Academic Press.

Rumelhart, D.E. (1980). Schemata: The building blocks of cognition. In R.J. Spiro, B.C. Bruce, & W.F. Brewer (Eds.) *Theoretical Issues in Reading Comprehension: Perspectives from Cognitive Psychology, Linguistics, Artificial Intelligence, and Education* (pp. 33 – 58). Hillsdale, NJ: Lawrence Erlbaum Associates.

Ryan, M. (1991). *Possible Worlds, Artificial Intelligence, and Narrative Theory*. Bloomington, IN: Indiana University Press.

Sacerdoti, E.D. (1977). *A Structure for Plans and Behavior*. New York: Elsevier.

Sadock, J.M. (1990). Comments on Vanderveken and on Cohen and Levesque. In P.R. Cohen, J. Morgan, & M.E. Pollack (Eds.) *Intentions in Communication* (pp. 257-270). Cambridge, MA: The MIT Press.

Schank, R.C. & Abelson, R.P. (1977). *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Seif El-Nasr, M. & Horswill, I. (2003). Real-time lighting design for interactive narrative. In *Proceedings of the 2nd International Conference on Virtual Storytelling*, 12-20.

Sengers, P. (2000a). Narrative Intelligence. In K. Dautenhahn (Ed.) *Human Cognition and Social Agent Technology* (pp. 1-26). Philadelphia: John Benjamins Publishing.

Sengers, P. (2000b). Schizophrenia and narrative in artificial agents. In *Proceedings of the 1st International Workshop on Narrative and Interactive Learning Environments*, 13-34.

Sgouros, N.M. (1999). Dynamic generation, management, and resolution of interactive plots. *Artificial Intelligence*, 107, 29-62.

Sharples, M. (1999). *How We Write: Writing as Creative Design*. London: Routledge.

Silva, A., Raimundo, G., & Paiva, A. (2003). Tell me that bit again... Bringing interactivity to a virtual storyteller. *Proceedings of the 2nd International Conference on Virtual Storytelling*, 146-154.

Smith, D.E. & Weld, D.S. (1998). Conformant graphplan. *Proceedings of the 15th National Conference on Artificial Intelligence*, 889-896.

Smith, T.C. & Witten, I.H. (1991). A planning mechanism for generating story texts. *Literary and Linguistic Computation*, 6(2), 119-126.

Sobral, D., Machado, I., & Paiva, A. (2003a). Managing authorship in plot conduction. *Proceedings of the 2nd International Conference on Virtual Storytelling*, 57-64.

Sobral, D., Machado, I., & Paiva, A. (2003b). Machiavellian characters and the edutaiment paradox. *Proceedings of the 4th International Conference on Intelligent Virtual Agents*, 333-340.

Theune, M., Faas, S., Nijholt, A., & Heylen, D. (2003). The virtual storyteller: Story creation by intelligent agents. *Proceedings of the 1st International Conference for Interactive Digital Storytelling and Entertainment*, 204-215.

Thomas, F. & Johnson, O. (1981). *Disney Animation: The Illusion of Life*. New York: Abbeville Press.

Thorndyke, P.W. (1977). Cognitive structures in comprehension and memory of narrative discourse. *Cognitive Psychology*, 9, 77-110.

Thorndyke, P.W. & Yekovich, F.R. (1980). A critique of schema-based theories of human story memory. *Poetics*, 9, 23-49.

Trabasso, T. & Sperry, L.L. (1985). Causal relatedness and importance of story events. *Journal of Memory and Language*, 24, 595-611.

van den Broek, P. (1988). The effects of causal relations and hierarchical position on the importance of story statements. *Journal of Memory and Language*, 27, 1-22.

Weld, D.S. (1994). An introduction to least commitment planning. *AI Magazine*, 15(4), 27-61.

Wilensky, R. (1981). Meta-planning: Representing and using knowledge about planning in problem solving and natural language understanding. *Cognitive Science*, 5, 197-233.

Wilensky, R. (1982). Points: A theory of the structure of stories in memory. In W.G. Lehnert & M.H. Ringle (Eds.) *Strategies for Natural Language Processing* (pp. 345-374). Hillsdale, NJ: Lawrence Erlbaum Associates.

Young, R.M. & Moore, J.D. (1994). DPOCL: A principled approach to discourse planning. *Proceedings of the 7th International Workshop on Text Generation*, 13-20.

Young, R.M., Moore, J.D., & Pollack, M.E. (1994). Towards a principled representation of discourse plans. *Proceedings of the 16th Annual Meeting of the Cognitive Science Society*, 946-951.

Young, R.M., Pollack, M.E., & Moore, J.D. (1994). Decomposition and causality in partial-order planning. *Proceedings of the 2nd International Conference on Artificial Intelligence and Planning Systems*. 188-193.

Young, R.M. & Riedl, M.O. (2003). Towards an architecture for intelligent control of narrative in interactive virtual worlds. *Proceedings of the 2003 International Conference on Intelligent User Interfaces*, 310-312.

Young, R.M., Riedl, M.O., Branly, M., Jhala, A.H., Martin, R.J., & Saretto, C.J. (2004). An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development*, 1(1), 51-70.

Zyda, M., Hiles, J., Mayberry, A., Wardynski, C., Capps, M., Osborn, B., Russell, S., Robaszewski, M., & Davis, M. (2003). Entertainment R&D for defense. *IEEE Computer Graphics and Applications*, January/Feburary 2003, 2-10.

Appendix A

Evaluation Materials

A.1. Plot Coherence Evaluation Study

The subjects in the plot coherence study were given a pen-and-paper questionnaire to fill out. The trial operated in three phases. In the first phase, the subjects read the instructions and were given an opportunity to ask questions. The instruction page is shown in Figure A.1. In the second phase, the subjects read a story containing both generated and hand-authored narrative. The story page is shown in Figure A.2. Note that it is impossible to distinguish generated parts of the story from hand-authored parts of the story through style or visual cues alone. In the third phase, the subjects rated the importance of sentences from the story and circled the most appropriate rating on a four-point Likert scale. The first page of the questionnaire is shown in Figure A.3. Each sentence in the story has a line in the questionnaire.

Instructions

You are asked to read a short story. Afterward, you will be asked to rank the importance of each sentence to the outcome of the story. A sentence can be "not very important", "somewhat not important", "somewhat important", or "very important". There is no right or wrong answer and you are not being evaluated. You may wish to read the story several times before you begin, although you may refer back to the story as often as you wish. You are not being timed.

If you have any questions at this time, please ask.

Figure A.1. Instruction sheet for questionnaire in the plot coherence evaluation study.

Story

There is a woman named Jasmine. There is a king named Mamoud. This is a story about how King Mamoud becomes married to Jasmine.

There is a hoard of gold. There is a dragon. The dragon has the hoard of gold. There is a magic lamp. The dragon has the magic lamp. There is a magic genie. The genie is confined within the magic lamp.

King Mamoud is not married. Jasmine is very beautiful. King Mamoud sees Jasmine and instantly falls in love with her. King Mamoud wants to marry Jasmine. There is a brave knight named Aladdin. Aladdin is loyal to the death to King Mamoud. King Mamoud orders Aladdin to get the magic lamp for him. Aladdin wants King Mamoud to have the magic lamp. Aladdin travels from the castle to the mountains. Aladdin slays the dragon. The dragon is dead. Aladdin takes the magic lamp from the dead body of the dragon. There is a peasant named Ali. Ali is in the mountains. Ali is very poor. Ali begs Aladdin for some money. Aladdin is very generous. Aladdin wants to help Ali. Aladdin takes the hoard of gold from the dead body of the dragon. Aladdin gives the hoard of gold to Ali. Aladdin travels from the mountains to the castle. Aladdin hands the magic lamp to King Mamoud. The genie is in the magic lamp. King Mamoud rubs the magic lamp and summons the genie out of it. The genie is not confined within the magic lamp. King Mamoud controls the genie with the magic lamp. King Mamoud uses the magic lamp to command the genie to make Jasmine love him. The genie wants Jasmine to be in love with King Mamoud. The genie casts a spell on Jasmine making her fall in love with King Mamoud. Jasmine is madly in love with King Mamoud. Jasmine wants to marry King Mamoud. There is a court jester. The court jester heard about the wedding. The court jester wanted everyone to know about the wedding. There is a town near the castle. The court jester went to the town. The court jester proclaimed the marriage of King Mamoud to Jasmine. The King Mamoud and Jasmine wed in an extravagant ceremony.

King Mamoud and Jasmine are married. The end.

Figure A.2. Story sheet for questionnaire in the plot coherence study.

Questions for the Story

	Not at all important	Somewhat not important	Somewhat important	Very important
The court jester went to the town	1	2	3	4
Ali begs Aladdin for some money	1	2	3	4
The court jester wanted everyone to know about the wedding	1	2	3	4
There is a town near the castle	1	2	3	4
King Mamoud sees Jasmine and instantly falls in lover with her	1	2	3	4
King Mamoud rubs the magic lamp and summons the genie out of it	1	2	3	4
Ali is in the mountains	1	2	3	4
King Mamoud is not married	1	2	3	4
The dragon has the magic lamp	1	2	3	4
Jasmine wants to marry King Mamoud	1	2	3	4
There is a hoard of gold	1	2	3	4
There is a brave knight named Aladdin	1	2	3	4
Jasmine is madly in love with King Mamoud	1	2	3	4
King Mamoud controls the genie with the magic lamp	1	2	3	4
Aladdin wants to help Ali	1	2	3	4
The genie is in the magic lamp	1	2	3	4
Ali is very poor	1	2	3	4
There is a woman named Jasmine	1	2	3	4
There is a magic genie	1	2	3	4
Aladdin takes the magic lamp from the dead body of the dragon	1	2	3	4
King Mamoud orders Aladdin to get the magic lamp for him	1	2	3	4

Figure A.3. First page of questions in the plot coherence study.

A.2. Character Believability Evaluation Study

The subjects in the plot coherence study were given a pen-and-paper questionnaire to fill out. Subjects were randomly assigned to one of two conditions: the test condition and the control condition. The trial process was identical for both groups, but each group received a different story and, consequently, a different set of question/answer pairs to rate, although there were many similarities between stories. The trial operated in three phases. In the first phase, the subjects of both conditions read the same set of instructions and were given an opportunity to ask questions. The instruction page is shown in Figure A.4. The instructions included two same question/answer pairs to familiarize the subjects with the task they will be performing since rating goodness of answers is not a typical task. For the remainder of the trial, subjects in each condition received distinct materials. In the second phase, the subjects read a story containing both generated and hand-authored narrative. The story page for the test condition is shown in Figure A.5. In the third phase, the subjects rated the goodness of answers in question/answer pairs. The first page of the questionnaire for the test condition is shown in Figure A.6. The story page for the control condition is shown in Figure A.7. Note that the story for the control condition is shorter than the story for the test condition, although there are many similarities between both stories. The first page of the questionnaire for the control condition is shown in Figure A.8. Since the story for the control condition is shorter than the story for the test condition, the number of question/answer pairs in the control condition is necessarily fewer. However, the question/answer pairs identical in both conditions occur in the same order in both questionnaires to partially control for unintended ordering affects.

Instructions

You are asked to read a short story. Afterward, you will be presented with a series of questions about the story. Instead of answering each question, an answer will be given for each question. You are asked to rate whether the given answer is a good answer to that question or a bad answer to that question by selecting one of four options: "very bad answer", "somewhat bad answer", "somewhat good answer", or "very good answer". For example, read the story below.

There is a boy named Jack. Jack ran out of water at the house. Jack went up the hill. Jack fetched a pail of water. Jack fell down and broke his crown.

Questions and corresponding answers will be provided, such as the following. You do not need to rate the example questions.

Q: Why did Jack go up the hill?
A: Because Jack wanted to fetch a pail of water.

Very bad answer	*Somewhat bad answer*	*Somewhat good answer*	*Very good answer*

Q: Why did Jack go up the hill?
A: Because Jack ran out of water at the house.

Very bad answer	*Somewhat bad answer*	*Somewhat good answer*	*Very good answer*

Q: Why did Jack go up the hill?
A: Because Jack wanted to fall down and break his crown.

Very bad answer	*Somewhat bad answer*	*Somewhat good answer*	*Very good answer*

You do not need to answer the questions, merely select the rating that most appropriately describes the quality of the answer for each question. You will see the same question several times, but each time it will be paired with a slightly different answer. There is no right or wrong answer and you are not being evaluated. You may wish to read the story several times before you begin, although you can refer back to the story as often as you wish. You may refer back to the story at any time. You are not being timed.

If you have any questions at this time, please ask.

Figure A.4. Instruction sheet for questionnaire in both conditions of the character believability evaluation study.

Story

There is a woman named Jasmine. There is a king named Mamoud. This is a story about how King Mamoud becomes married to Jasmine. There is a magic genie. This is also a story about how the genie dies.

There is a magic lamp. There is a dragon. The dragon has the magic lamp. The genie is confined within the magic lamp.

King Mamoud is not married. Jasmine is very beautiful. King Mamoud sees Jasmine and instantly falls in love with her. King Mamoud wants to marry Jasmine. There is a brave knight named Aladdin. Aladdin is loyal to the death to King Mamoud. King Mamoud orders Aladdin to get the magic lamp for him. Aladdin wants King Mamoud to have the magic lamp. Aladdin travels from the castle to the mountains. Aladdin slays the dragon. The dragon is dead. Aladdin takes the magic lamp from the dead body of the dragon. Aladdin travels from the mountains to the castle. Aladdin hands the magic lamp to King Mamoud. The genie is in the magic lamp. King Mamoud rubs the magic lamp and summons the genie out of it. The genie is not confined within the magic lamp. King Mamoud controls the genie with the magic lamp. King Mamoud uses the magic lamp to command the genie to make Jasmine love him. The genie wants Jasmine to be in love with King Mamoud. The genie casts a spell on Jasmine making her fall in love with King Mamoud. Jasmine is madly in love with King Mamoud. Jasmine wants to marry King Mamoud. The genie has a frightening appearance. The genie appears threatening to Aladdin. Aladdin wants the genie to die. Aladdin slays the genie. King Mamoud and Jasmine wed in an extravagant ceremony.

The genie is dead. King Mamoud and Jasmine are married. The end.

Figure A.5. Story sheet for questionnaire in the test condition of the character believability study.

Questions for the Story

1.
Q: Why did Aladdin travel from the mountains to the castle?
A: Because the genie wanted to cast a spell on Jasmine, making her fall in love with King Mamoud.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

2.
Q: Why did King Mamoud rub the magic lamp and summon the genie out of it?
A: Because King Mamoud wanted to marry Jasmine.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

3.
Q: Why did Aladdin slay the dragon?
A: Because King Mamoud ordered Aladdin to get the magic lamp for him.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

4.
Q: Why did Aladdin travel from the castle to the mountains?
A: Because Aladdin wanted to travel from the mountains to the castle.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

5.
Q: Why did the genie cast a spell on Jasmine, making her fall in love with King Mamoud?
A: Because King Mamoud commanded the genie to make Jasmine love him.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

6.
Q: Why did Aladdin travel from the mountains to the castle?
A: Because Aladdin wanted to hand the magic lamp to King Mamoud.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

7.
Q: Why did King Mamoud order Aladdin to get the magic lamp for him?
A: Because Aladdin wanted to slay the genie.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

Figure A.6. First page of questions in the test condition of the character believability study.

Story

There is a woman named Jasmine. There is a king named Mamoud. This is a story about how King Mamoud becomes married to Jasmine. There is a magic genie. This is also a story about how the genie dies.

There is a magic lamp. There is a dragon. The dragon has the magic lamp. The genie is confined within the magic lamp.

There is a brave knight named Aladdin. Aladdin travels from the castle to the mountains. Aladdin slays the dragon. The dragon is dead. Aladdin takes the magic lamp from the dead body of the dragon. Aladdin travels from the mountains to the castle. Aladdin hands the magic lamp to King Mamoud. The genie is in the magic lamp. King Mamoud rubs the magic lamp and summons the genie out of it. The genie is not confined within the magic lamp. The genie casts a spell on Jasmine making her fall in love with King Mamoud. Jasmine is madly in love with King Mamoud. Aladdin slays the genie. King Mamoud and Jasmine wed in an extravagant ceremony.

The genie is dead. King Mamoud and Jasmine are married. The end.

Figure A.7. Story sheet for questionnaire in the control condition of the character believability study.

Questions for the Story

1.
Q: Why did Aladdin travel from the mountains to the castle?
A: Because the genie wanted to cast a spell on Jasmine, making her fall in love with King Mamoud.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

2.
Q: Why did King Mamoud rub the magic lamp and summon the genie out of it?
A: Because King Mamoud wanted to marry Jasmine.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

3.
Q: Why did Aladdin travel from the castle to the mountains?
A: Because Aladdin wanted to travel from the mountains to the castle.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

4.
Q: Why did Aladdin travel from the mountains to the castle?
A: Because Aladdin wanted to hand the magic lamp to King Mamoud.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

5.
Q: Why did the genie cast a spell on Jasmine, making her fall in love with King Mamoud?
A: Because King Mamoud wanted to marry Jasmine.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

6.
Q: Why did Aladdin slay the dragon?
A: Because Aladdin wanted to slay the genie.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

7.
Q: Why did Aladdin travel from the castle to the mountains?
A: Because Aladdin wanted to hand the magic lamp to King Mamoud.

Very bad answer	Somewhat bad answer	Somewhat good answer	Very good answer

Figure A.8. First page of questions in the control condition of the character believability study.

Appendix B

Example Story World Domain Specifications

Fabulist takes several different inputs:

- A description of the initial state of the world,
- A description of the final state of the world (story goal state),
- The undetermined set,
- Mutex sets,
- Author goals,
- A fabula action schema library,
- A fabula planning heuristic function,
- A personality model definition,
- A discourse action library,
- A discourse planning heuristic function,
- And a set of natural language templates (in the typical configuration using a template-based text media realizer).

The fabula planner component of Fabulist alone requires the first eight. The undetermined set, mutex sets, author-goals, and personality model are optional; the fabula planner does not need to have indeterminism in the initial state of the story world. If the fabula planning heuristic of discourse planning heuristic is not specified, Fabulist uses a default heuristic function. However, it is never beneficial to use the default heuristic function. The initial state, goal state, undetermined set, mutex sets, author goals, and fabula heuristic function are specified in a single *problem statement*. The fabula action schema library is provided separately from the problem statement so that different problems (corresponding to different stories) can be told using the same action schema library. This appendix gives the problem specifications (minus the heuristic function) and action libraries used to generate the stories used as examples and in the evaluation study in this dissertation.

B.1. Bribery

The Bribery story is used in the examples in Sections 4.5 and 5.3.4. The story generated by Fabulist is about how the President becomes corrupt. That is, the President is not corrupt in the initial state of the story world but is corrupt in the final state of the story world. The example in Section 4.5 motivates how a story planner that reasons about character intentions forms frames of commitment and find motivating actions. The example is Section 5.3.4 demonstrates how recommendations work and how sophisticated heuristics are needed to prevent the planner from making errors in judgment about a character's traits in cases where one character is operating under the will of another.

Table B.1. Initialization parameters for the Bribery story problem.

Initial State	Goal State
(person president) (person villain) (person hero) (thing money1) (money money1) (bank bank1) (at money1 bank1) (evil villain) (lawful hero) (intends villain (controls villain president))	(corrupt president)

Table B.2. Action library for the Bribery story problem.

```
(define (action bribe)
   :parameters (?person ?target ?money)
   :actors (?person)
   :constraints ((person ?person) (person ?target) (money ?money))
   :precondition ((has ?person ?money) (:neq ?person ?target))
   :recommendation ((evil ?person))
   :effect ((corrupt ?target) (controls ?person ?target) (has ?target ?money)
            (:not (has ?person ?money))))

(define (action give)
   :parameters (?person ?target ?thing)
   :actors (?person)
   :constraints ((person ?person) (person ?target) (thing ?thing))
   :precondition ((has ?person ?thing) (:neq ?person ?target))
   :effect ((has ?target ?thing) (:not (has ?person ?thing))))

(define (action coerce)
   :parameters (?person ?victim ?objective)
   :actors (?person)
   :constraints ((person ?person) (person ?victim))
   :precondition ((afraid-of ?victim ?person) (:neq ?person ?victim))
   :recommendation ((evil ?person))
   :effect ((intends ?victim ?objective)))

(define (action threaten)
   :parameters (?person ?victim)
   :actors (?person)
   :constraints ((person ?person) (person ?victim))
   :precondition nil
   :recommendation ((evil ?person))
   :effect ((afraid-of ?victim ?person)))

(define (action steal)
   :parameters (?person ?money ?bank)
   :actors (?person)
   :constraints ((person ?person) (money ?money) (bank ?bank))
   :precondition ((at ?money ?bank))
   :recommendation ((:not (lawful ?person)))
   :effect ((has ?person ?money) (:not (at ?money ?bank))))
```

B.2. Deer Hunter

The Deer Hunter story demonstrates how character actions can be part of multiple intentions, as described in Section 4.4.3.1. In the story the only character, Bubba, robs a bank and shoots a deer. The actions of picking up and loading a shotgun can be part of the character's intention to get money or his intention to not be hungry. The fabula planner generates story plans for all possible combinations of single and overloaded intentions. Whether or not the actions of picking up and loading the shotgun are part of both intentions or not affects the valid orderings for motivating actions.

Two of the actions in the action library, `Decide-to-eat` and `Decide-to-get-money` are happenings and consequently do not need to be part of any frame of commitment in order for a plan to be considered complete. The actions are marked as special by setting the slot `:not-needs-intention` to true.

Table B.3. Initialization parameters for the Deer Hunter story problem.

Initial State	Goal State
(person bubba) (mobile-person bubba) (person clerk) (animal bambi) (thing shotgun1) (gun shotgun1) (thing ammo1) (ammo ammo1) (thing money1) (money money1) (place forest) (place house) (place bank) (at bambi forest) (at bubba house) (at shotgun1 house) (at ammo1 house) (at clerk bank) (has clerk money1) (ammo-for ammo1 shotgun1) (hungry bubba) (alive bubba) (alive clerk) (alive bambi) (path house bank) (path bank forest) (path house forest) (path forest bank)	(has bubba money1) (:not (hungry bubba))

Table B.4. Action library for the Deer Hunter story problem.

```
(define (action go)
   :parameters (?mover ?dest ?from)
   :actors (?mover)
   :constraints ((mobile-person ?mover) (place ?dest) (place ?from))
   :precondition ((at ?mover ?from) (path ?from ?dest) (:neq ?dest ?from))
   :effect ((at ?mover ?dest) (:not (at ?mover ?from))))

(define (action shoot)
   :parameters (?shooter ?target ?gun ?place)
   :actors (?shooter)
   :constraints ((person ?shooter) (animal ?target) (gun ?gun) (place ?place))
   :precondition ((at ?shooter ?place) (at ?target ?place) (has ?shooter ?gun)
                  (loaded ?gun))
   :effect ((:not (alive ?target))))

(define (action eat)
   :parameters (?eater ?food ?place)
   :actors (?eater)
   :constraints ((person ?eater) (animal ?food) (place ?place))
   :precondition ((at ?eater ?place) (at ?food ?place) (:not (alive ?food)))
   :effect ((:not (hungry ?eater))))

(define (action pick-up)
   :parameters (?person ?thing ?place)
   :actors (?person)
   :constraints ((person ?person) (thing ?thing) (place ?place))
   :precondition ((at ?person ?place) (at ?thing ?place))
   :effect ((has ?person ?thing) (:not (at ?thing ?place))))

(define (action steal)
   :parameters (?thief ?victim ?thing ?gun ?place)
   :actors (?thief)
   :constraints ((person ?thief) (person ?victim) (thing ?thing) (gun ?gun) (place ?place))
   :precondition ((at ?thief ?place) (at ?victim ?place) (has ?victim ?thing)
                  (has ?thief ?gun) (loaded ?gun) (:neq ?thief ?victim))
   :effect ((has ?thief ?thing) (:not (has ?victim ?thing))))

(define (action load)
   :parameters (?person ?gun ?ammo)
   :actors (?person)
   :constraints ((person ?person) (gun ?gun) (ammo ?ammo))
   :precondition ((has ?person ?gun) (has ?person ?ammo) (ammo-for ?ammo ?gun))
   :effect ((loaded ?gun) (in ?ammo ?gun) (:not (has ?person ?ammo))))

(define (action decide-to-eat)
   :parameters (?person)
   :actors (?person)
   :constraints ((person ?person))
   :precondition ((hungry ?person))
   :not-needs-intention t
   :effect ((intends ?person (:not (hungry ?person)))))

(define (action decide-to-get-money)
   :parameters (?person ?money1)
   :actors (?person)
   :constraints ((person ?person) (money ?money))
   :precondition ((:not (has ?person ?money)))
   :not-needs-intention t
   :effect ((intends ?person (has ?person ?money))))
```

B.3. Secret Agent

The Secret Agent story is used in the example in Section 6.3. The story demonstrates the initial state revision planning algorithm by setting up a problem for the fabula planner that cannot be solved without the ability to modify the description of the initial state of the story world. The story world has two characters: a terrorist mastermind and a secret agent. The outcome of the story is that the terrorist mastermind is dead. However, the mastermind starts out in the safety of his fortress which is guarded so that no person carrying a weapon can enter the building. If the only gun in the story world is outside of the fortress, the fabula planner cannot find a solution to the problem. However, if the location of a gun is undetermined in the initial state, then an ISR planner can determine that the gun needs to be inside the fortress so that the secret agent can enter the fortress, pick up the gun, and use it to assassinate the terrorist mastermind.

There are two versions of the initialization parameters given here. The first version sets up a problem that cannot be solved because the only weapon is outside of the fortress. The second version is identical to the first except that the location of the gun is undetermined in the initial state. Both versions use the same action library.

Table B.5. Initialization parameters for the Secret Agent story problem that are guaranteed to fail.

Initial State	Goal State
(place headquarters) (place dropbox) (place lobby) (place office) (place cache) (place courtyard) (connection headquarters dropbox) (connection headquarters courtyard) (connection dropbox courtyard) (connection lobby cache) (connection lobby office) (connection cache office) (guarded-connection courtyard lobby) (papers dox) (at dox dropbox) (papers-required courtyard lobby dox) (person secret-agent) (mobile secret-agent) (at secret-agent headquarters) (alive secret-agent) (person mastermind) (at mastermind office) (weapon gun) (at gun headquarters) (loaded gun) (intends secret-agent (:not (alive mastermind)))	(:not (alive mastermind))

Table B.6. Initialization parameters for the Secret Agent story problem that uses initial state revision.

<table>
<tr><th>Initial State</th><th>Goal State</th></tr>
<tr><td rowspan="5">(place headquarters)
(place dropbox)
(place lobby)
(place office)
(place cache)
(place courtyard)
(connection headquarters dropbox)
(connection headquarters courtyard)
(connection dropbox courtyard)
(connection lobby cache)
(connection lobby office)
(connection cache office)
(guarded-connection courtyard lobby)
(papers dox)
(at dox dropbox)
(papers-required courtyard lobby dox)
(person secret-agent)
(mobile secret-agent)
(at secret-agent headquarters)
(alive secret-agent)
(person mastermind)
(at mastermind office)
(weapon gun)
(intends secret-agent (:not (alive mastermind)))</td><td>(:not (alive mastermind))</td></tr>
<tr><th>Undetermined Set</th></tr>
<tr><td>(at gun ?place)
(loaded gun)</td></tr>
<tr><th>Mutex Sets</th></tr>
<tr><td>((at gun ?place))</td></tr>
</table>

Table B.7. Action library for the Secret Agent story problem.

```
(define (action move)
   :parameters (?person ?start ?dest)
   :actors (?person)
   :constraints ((person ?person) (place ?start) (place ?dest))
   :precondition ((at ?person ?start) (mobile ?person) (alive ?person)
                  (connection ?start ?dest) (:neq ?start ?dest))
   :effect ((at ?person ?dest) (:not (at ?person ?start))))

(define (action move-through-guards)
   :parameters (?person ?start ?dest ?papers)
   :actors (?person)
   :constraints ((person ?person) (place ?start) (place ?dest) (papers ?papers))
   :precondition ((at ?person ?start) (mobile ?person) (alive ?person)
                  (:not (armed ?person)) (has ?person ?papers) (:neq ?start ?dest)
                  (guarded-connection ?start ?dest) (papers-required ?start ?dest ?papers))
   :effect ((at ?person ?dest) (:not (at ?person ?start))))

(define (action kill)
   :parameters (?person ?victim ?place ?weapon)
   :actors (?person)
   :constraints ((person ?person) (person ?victim) (place ?place) (weapon ?weapon))
   :precondition ((at ?person ?place) (at ?victim ?place) (armed ?person)
                  (has ?person ?weapon) (loaded ?weapon) (:neq ?person ?victim))
   :effect ((:not (alive ?victim))))

(define (action pickup)
   :parameters (?person ?thing ?place)
   :actors (?person)
   :constraints ((person ?person) (thing ?thing) (place ?place))
   :precondition ((at ?person ?place) (at ?thing ?place))
   :effect ((has ?person ?thing) (:not (at ?thing ?place))))

(define (action pickup-weapon)
   :parameters (?person ?weapon ?place)
   :actors (?person)
   :constraints ((person ?person) (weapon ?weapon) (place ?place))
   :precondition ((at ?person ?place) (at ?weapon ?place))
   :effect ((has ?person ?weapon) (armed ?person) (:not (at ?weapon ?place))))
```

B.4. Aladdin

The Aladdin story is used in the evaluation studies described in Chapter 9. The characters in the story world commit to one or more different goals at different times. The story is more sophisticated than the previous examples in that the actions of the characters motivate other characters to commit to goals. Additionally, the action library contains schemata for actions that require joint character intentions. Specifically, the `Marry` action requires the characters in the role of bride and groom to be acting intentionally. There are two action schemata of happenings in the action library: `Fall-in-love` and `Appear-threatening`. Both actions are things that happen without necessarily being intentional.

The plot coherence evaluation study and character believability evaluation study use slightly different initialization configurations: the plot coherence evaluation has a simpler goal state. Both configurations are given below. In the character believability evaluation, two versions of the same story were generated, one using the IPOCL fabula planning algorithm, the other using a standard POCL planning algorithm. The initialization configuration and action library were identical for both versions. The complete action library used in both study conditions is shown in Tables B.10 and B.11. Table B.11 enumerates the action schemata that have character intentions as effects. While both conditions used identical action libraries, the conventional planner in the test condition did not use any of the motivating actions (except for `Fall-in-love`, in which case it used the operation for its other effects).

Table B.8. Initialization parameters for the Aladdin story problem used in the plot coherence evaluation study.

Initial State	Goal State
(character Aladdin) (male Aladdin) (knight Aladdin) (at Aladdin castle) (alive Aladdin) (single Aladdin) (loyal-to Aladdin Mamoud) (character King) (male Mamoud) (king Mamoud) (at Mamoud castle) (alive Mamoud) (single Mamoud) (character Jasmine) (female Jasmine) (at Jasmine castle) (alive Jasmine) (single Jasmine) (beautiful Jasmine) (character Dragon) (monster Dragon) (dragon Dragon) (at Dragon mountain) (alive Dragon) (scary Dragon) (character Genie) (monster Genie) (genie Genie) (in Genie lamp) (confined Genie) (alive Genie) (scary Genie) (place castle) (place mountain) (thing lamp) (magic-lamp lamp) (has Dragon lamp)	(married-to King Jasmine)

Table B.9. Initialization parameters for the Aladdin story problem used in the character believability evaluation study.

Initial State	Goal State
(character Aladdin) (male Aladdin) (knight Aladdin) (at Aladdin castle) (alive Aladdin) (single Aladdin) (loyal-to Aladdin Mamoud) (character King) (male Mamoud) (king Mamoud) (at Mamoud castle) (alive Mamoud) (single Mamoud) (character Jasmine) (female Jasmine) (at Jasmine castle) (alive Jasmine) (single Jasmine) (beautiful Jasmine) (character Dragon) (monster Dragon) (dragon Dragon) (at Dragon mountain) (alive Dragon) (scary Dragon) (character Genie) (monster Genie) (genie Genie) (in Genie lamp) (confined Genie) (alive Genie) (scary Genie) (place castle) (place mountain) (thing lamp) (magic-lamp lamp) (has Dragon lamp)	(married-to King Jasmine) (:not (alive Genie))

Table B.10. Part of the action library for the Aladdin story problem.

```
(define (action travel)
   :parameters (?traveller ?from ?dest)
   :actors (?traveller)
   :constraints ((character ?traveller) (place ?from) (place ?dest))
   :precondition ((at ?traveller ?from) (alive ?traveller) (:neq ?from ?dest))
   :effect ((at ?traveller ?dest) (:not (at ?traveller ?from))))

(define (action slay)
   :parameters (?slayer ?monster ?place)
   :actors (?slayer)
   :constraints ((character ?slayer) (monster ?monster) (place ?place))
   :precondition ((at ?slayer ?place) (at ?monster ?place) (alive ?slayer) (alive ?monster)
                  (:neq ?slayer ?monster))
   :effect ((:not (alive ?monster))))

(define (action pillage)
   :parameters (?pillager ?body ?thing ?place)
   :actors (?pillager)
   :constraints ((character ?pillager) (character ?body) (thing ?thing) (place ?place))
   :precondition ((at ?pillager ?place) (at ?body ?place) (has ?body ?thing)
                  (:not (alive ?body)) (alive ?pillager) (:neq ?pillager ?body))
   :effect ((has ?pillager ?thing) (:not (has ?body ?thing))))

(define (action give)
   :parameters (?giver ?givee ?thing ?place)
   :actors (?giver)
   :constraints ((character ?giver) (character ?givee) (thing ?thing) (place ?place))
   :precondition ((at ?giver ?place) (at ?givee ?place) (has ?giver ?thing) (alive ?giver)
                  (alive ?givee) (:neq ?giver ?givee))
   :effect ((has ?givee ?thing) (:not (has ?giver ?thing))))

(define (action summon)
   :parameters (?char ?genie ?lamp ?place)
   :actors (?char)
   :constraints ((character ?char) (genie ?genie) (magic-lamp ?lamp) (place ?place))
   :precondition ((at ?char ?place) (has ?char ?lamp) (in ?genie ?lamp) (alive ?char)
                  (alive ?genie))
   :effect ((at ?genie ?place) (:not (in ?genie ?lamp)) (:not (confined ?genie))
            (controls ?char ?genie ?lamp)))

(define (action love-spell)
   :parameters (?genie ?target ?lover)
   :actors (?genie)
   :constraints ((genie ?genie) (character ?target) (character ?lover))
   :precondition ((:not (confined ?genie)) (:not (loves ?target ?lover)) (alive ?genie)
                  (alive ?target) (alive ?lover) (:neq ?genie ?target) (:neq ?genie ?lover)
                  (:neq ?target ?lover))
   :effect ((loves ?target ?lover) (intends ?target (married-to ?target ?lover))))

(define (action marry)
   :parameters (?groom ?bride ?place)
   :actors (?groom ?bride)
   :constraints ((male ?groom) (female ?bride) (place ?place))
   :precondition ((at ?groom ?place) (at ?bride ?place) (loves ?groom ?bride)
                  (loves ?bride ?groom) (alive ?groom) (alive ?bride))
   :effect ((married-to ?groom ?bride) (married-to ?bride ?groom)))
```

Table B.11. Motivating actions for the Aladdin story problem.

```
(define (action fall-in-love)
   :parameters (?male ?female ?place)
   :actors (?male)
   :constraints ((male ?male) (female ?female) (place ?place))
   :precondition ((at ?male ?place) (at ?female ?place) (single ?male) (alive ?male)
                  (alive ?female) (beautiful ?female) (:not (loves ?female ?male))
                  (:not (loves ?male ?female)))
   :not-needs-intention t
   :effect ((loves ?male ?female) (intends ?male (married-to ?male ?female))))

(define (action order)
   :parameters (?king ?knight ?place ?objective)
   :actors (?king)
   :constraints ((king ?king) (knight ?knight) (place ?place))
   :precondition ((at ?king ?place) (at ?knight ?place) (alive ?king) (alive ?knight)
                  (loyal-to ?knight ?king))
   :effect ((intends ?knight ?objective)))

(define (action command)
   :parameters (?char ?genie ?lamp ?objective)
   :actors (?char)
   :constraints ((character ?char) (genie ?genie) (magic-lamp ?lamp))
   :precondition ((has ?char ?lamp) (controls ?char ?genie ?lamp) (alive ?char)
                  (alive ?genie) (:neq ?char ?genie))
   :effect ((intends ?genie ?objective)))

(define (action appear-threatening)
   :parameters (?monster ?char ?place)
   :actors (?monster)
   :constraints ((monster ?monster) (character ?char) (place ?place))
   :precondition ((at ?monster ?place) (at ?char ?place) (scary ?monster)
                  (:neq ?monster ?char))
   :not-needs-intention t
   :effect ((intends ?char (:not (alive ?monster)))))
```