ABSTRACT

NIEHAUS, JAMES M. Cognitive Models of Discourse Comprehension for Narrative Generation. (Under the direction of Professor R. Michael Young).

Recent work in the area of narrative generation has sought to develop systems that automatically produce experiences for a user that are understood as stories. Much of this prior work, however, has focused on the structural aspects of narrative rather than the process of narrative comprehension undertaken by readers. Cognitive theories of narrative discourse comprehension define explicit models of a reader's mental state during reading. These cognitive models are created to test hypotheses and explain empirical results about the comprehension processes of readers. They do not often contain sufficient precision for implementation on a computer, and thus, they are not yet suitable for computational generation purposes. This dissertation employs cognitive models of narrative discourse comprehension to define an explicit computational model of a reader's comprehension process during reading, predicting aspects of narrative focus and inferencing with precision. This computational model is employed in a narrative discourse generation system to select content from an event log, creating discourses that satisfy comprehension criteria. The results of three experiments are presented and discussed, exhibiting empirical support for the computational reader model and the results of generation. This dissertation makes a number of contributions that advance the state-of-the-art in narrative discourse generation: a formal model of narrative focus, a formal model of online inferencing in narrative, a method of selecting narrative discourse content to satisfy comprehension criteria, and implementation and evaluation of these models.

Cognitive Models of Discourse Comprehension for Narrative Generation

by
James Niehaus

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

Computer Science

Raleigh, North Carolina

2009

APPROVED BY:

Dr. Robert St. Amant	Dr. Jon Doyle
Dr. Stephen Mitroff	Dr. James Lester
Dr. R. Michael Young Chair of Advisory Committee	

DEDICATION

To my family.

BIOGRAPHY

James Michael Niehaus was born in Plymouth, MI, but spent much of his childhood in South Carolina developing an interest in computers, video games, and science. James received his B.S. in Computer Science at the College of Charleston, completing two Bachelor's Essays and graduating in the Honors College. Immediately following, he attended graduate school at North Carolina State University and promptly joined the Liquid Narrative research group to study interactive narrative, games, and how virtual worlds are constructed for people.

ACKNOWLEDGMENTS

First and foremost, I would like to thank the members of my commitee, Robert St. Amant, Stephen Mitroff, Jon Doyle, and James Lester for their concern, support, and guidance.

During the course of my graduate and undergraduate careers, I have had the pleasure of working with several advisors whom have made an immense impact upon my growth as a professional. At the College of Charleston, Paul Buhler helped to usher me to graduate school, and Allan Strand and Jerry Boetje offered me my first taste of research. Mark Riedl, whom I worked with at Georgia Tech, has provided critical advice and support during our time together. Michael Young, my dissertation advisor and mentor for five years, has been an invaluable source of sage advice and an indefatigable supporter of my research. I was always better off for having spoken with Michael.

There are many who have provided comraderie and support over the last five years. A number of faculty at NC State University have assisted me in a variety of capacities: beyond my commitee, this list includes Len Anetta, Carol Miller, Robert Rodman, Munindar Singh, David Thuente, Annie Anton, and Mladen Vouk. I am proud to have been a memeber of Liquid Narrative group which has brought me through the process of the dissertation; I would like to thank Arnav Jhala, Leo Bae, Yuna Cheong, Jim Thomas, Tommy Vernieri, Joe Winegarden, Sam Munilla, Oliver Gray, Pat Cash, Joe France, and the many others. Fellow students Chris Hazard, Prachi Gauriar, Travis Breaux, and Paul Breimyer deserve special mention for many lively discussions and much support.

I would also like to thank my parents, Daniel and Janet Niehaus, for their love and support, and my brothers, Kevin and Joseph, for being a constant source of inspiration.

And Ashley, for everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

Consider the narrative in Figure 1.1; it is a story about vengeance for the slaying of a parent, a classical narrative theme. It contains a short introduction and motivating event, a difficulty that is overcome by the main character, and a climax and resolution to the initial plot. Yet, this narrative may seem unsatisfying. The middle portion about cooking does not seem to have anything to do with the introduction, and the narrative skips from one plot line to the other then back to the first. The narrative lacks cohesion, its elements do not seem to fit together well, and, worse, it is seemingly incoherent, the point or unifying theme is lost. Why should the reader care about the cooking segment if the introduction and resolution are both about a largely unrelated slaying?

Contrast the first narrative with the revised narrative in Figure 1.2. Here, three sentences have been added in the middle section that relate the cooking to the introduction and resolution. The causal structure of the earlier elements has not changed, but this narrative promotes a different, and perhaps more desirable, process of understanding by the reader. The first addition, "Jacob sees Cain at the cooking class.", reminds the reader of the slaying and Jacob's possible motivation for revenge. The second addition, "Jacob sees Cain at the party" reminds the reader yet again, and the third addition, "When it is time to cut the cake, Cain slowly hands the knife to Jacob." gives Jacob not only the motive but the means to take his revenge. The reader may now believe that the vengeance is at hand. Unlike the first narrative, the reader is reminded of the initiating event, the slaying of Tom by Cain, during the middle portion, and the reader is also presented with

1. Tom is Jacob's father.
2. Cain slays Tom during a robbery.
3. Jacob is lonely and wishes to have a party, but
4. he does not know how to cook.
5. Jacob signs up for a cooking class to learn how to cook.
6. He throws a big party at his house.
7. Jacob meets Cain late at night in an alley
8. Jacob slays Cain to avenge Tom's death.

Figure 1.1: Example narrative.

1. Tom is Jacob's father.
2. Cain slays Tom during a robbery.
3. Jacob is lonely and wishes to have a party, but
4. he does not know how to cook.
5. Jacob signs up for a cooking class to learn how to cook.
6. **Jacob sees Cain at the cooking class. ****
7. Jacob throws a big party at his house.
8. **Jacob sees Cain at the party. ****
9. **When it is time to cut the cake, Cain slowly hands the knife to Jacob. ****
10. Jacob meets Cain late at night in an alley.
11. Jacob slays Cain to avenge Tom's death.

Figure 1.2: A revised narrative. The asterisked lines have been inserted.

an opportunity to make a plausible prediction, that Jacob will slay Cain with the knife. By the additions, the reader's *focus* is maintained on the slaying of the father and the reader is prompted to *infer* the revenge.

Not all readers will make the prediction of revenge, be reminded of the initiating event, or prefer the second narrative to the first. However, research in cognitive models of narrative discourse comprehension [9, 10, 11, 12, 13, 14] and the experiments presented below indicate that many readers will experience these comprehension effects and the overall understanding of the narrative will be altered. Because of effects such as these, computational narrative systems, systems which understand or generate narrative, stand to benefit from models of narrative comprehension.

1.2 Computational Narrative Generation

Computational narrative generation is the process by which narratives are created by a computer. Narrative generation systems have been employed to create narratives for a variety of purposes and applications. Narrative generation has long interested Artificial Intelligence researchers because creating narratives is an ability that is unique to humans. However, generation systems have also found many practical uses, having been applied towards entertainment, training, and education.

Narrative generation systems may roughly be divided between *simulation or emergent systems*, those that primarily simulate the narrative world, and *deliberative systems* those that primarily deliberate over the choice of narrative elements and events. In simulation or emergent systems, the simulation is constructed and parameterized by the author in such a way that interesting narratives may emerge from the interactions between simulation elements [15]. Recent simulation systems attempt to create realistic world and character dynamics, and many employ complex models of characters [16, 17, 18, 19]. Deliberative systems, on the other hand, attempt to construct a narrative that satisfies criteria set forth by the author. Instead of allowing the simulation dynamics to drive the narrative from the bottom up, deliberative systems choose elements and events that satisfy goals and constraints.

The process of deliberative narrative generation begins with an author. The author creates the a representation of the dynamics of the story world, often by creating a set of events with logical preconditions and effects. The events describe what is possible in the story world. The author then creates a set of rules or heuristics that define which narratives are desirable and which are undesirable. Narrative rules may express classical plot structure, character dynamics, or even preferences over story situations. Together, the set of possible events and the narrative rules form a narrative generation problem. Solutions to the problem are narratives constructed from the events that both obey the story world dynamics and the narrative rules.

Much of the earlier work on deliberative narrative generation attempted to create story structure rules to impose structure onto sequences of events. This story structure may be contained in the definition of goals and subgoals [2], the evaluation functions of possible sequences of actions [20], schema based story fragments [21], or in the event preconditions and effects themselves [3]. Some recent systems have concentrated on evaluating the narrative experience from the reader's standpoint and creating story rules to optimize this experience. The narrative experience has been modeled on classical narratology principles [1], character believability [4], reader beliefs [5], suspense [22], and surprise [23].

Narrative is often decomposed into constituent pieces for generation [24, 4, 22]. A narrative may be divided between the *story*, *discourse*, and *medium*. The story (a.k.a. *fabula* or plot), is the representation of all of the characters, objects, setting, facts, and events that comprise the elements of the story world. The discourse, or *syuzhet*, is the telling of the story, it is a ordering over a subset of the story elements. In the story, a character may drink a cup of coffee before reading the newspaper and going to work. In the discourse, 'going to work' may be told before a flashback to the cup of coffee, and reading the newspaper may be left out entirely. Lastly, the medium is the translation of the elements in the discourse to a presentation format such as text, film, or comic book paneling.

One pipelining for complete narrative generation is to first decide upon the elements in the story, then choose from those elements and order them to create a discourse, and finally translate the discourse to a medium. The second task, creating the discourse from the story, is of particular interest when considering narrative comprehension. The appearance and ordering of elements is critical to the process of comprehension. Few murder mystery novels begin by giving the identity of the murderer, though the murders often happen in the beginning. On a lower level, interleaving the telling of events from ten different situations may severely reduce the ability of the reader to comprehend all of the situations.

The selection and ordering of content for narrative discourse is an interesting problem in its own right. In training simulations, After-Action Review (AAR) is used by instructors to highlight trainee actions and review the application strategy and tactics. A Simulation run may be treated as an extended story structure from which a discourse might be constructed. If the instructor wished to review a trainee's tactics in relation to a particular objective, the system might be directed to present the relevant events to the instructor in a coherent manner. In event-related data analysis tasks, such as using a set of news stories to explain how a politician was elected or performing intelligence analysis, narrative discourse generation may be used to combine information from otherwise disparate sources and create a comprehensible narrative answering the posed question.

A key question for generating narrative discourse is "What are the mechanisms by which the narrative will be comprehended?" To understand how people represent and comprehend narrative, cognitive models of narrative discourse comprehension may be employed.

1.3 Narrative Comprehension

Cognitive psychologists in the area of narrative discourse comprehension study how people understand and represent narrative. This area of research has identified many aspects of the narrative comprehension process supported by empirical study. These aspects include properties of mental models of narrative representation, narrative focus, and narrative inferencing.

There are multiple layers of discourse structure used in narrative comprehension [9, 10]. The separation of discourse structure into levels allows for a separation of the types of reasoning that occur at each level, differentiating the resolution of pronouns from the recognition of narrative themes. Each level uses its own structures to support reasoning, but the levels are not isolated from each other. Inferences at one level may improve understanding at higher or lower levels. Three of the most basic and universal levels are the Surface Code, the Textbase, and the Situation Model.

The surface code is the narrative as presented to the reader. In textual narratives, the surface code is the text itself. The textbase is the set of propositions that represent the meaning of the clauses in the surface code. The textbase is the foundation for the situation model, with which the present work is primarily concerned. The situation model is a mental representation of the narrative world state; including characters, settings, events in the plot and the diverse interrelationships between these items [9].

Narrative focus is the salience of an element in the reader's mind. Readers use recency of reading and the relatedness of narrative elements to aid in comprehension. Narrative elements that were recently mentioned are generally read faster and more easily comprehended, as are elements that are highly related to the current situation [11, 12, 13, 25]. These elements are the most salient in the reader's mind, and have the most focus. Narrative focus is often modeled by spreading activation through association nets, and a variety of methods have been employed to construct the association nets and spread the activation.

Narrative inferencing is the process by which a reader adds information not present in the narrative to his mental representation of the narrative. Readers may use inferences to resolve the referents of noun phrases and pronouns, character goals, determine themes, fill in gaps in causal chains [26], or predict the actions of characters [14]. For example, when reading that a character discovers a favorite potted plant has died during a vacation, the reader may believe that 1) the plant died from lack of watering and 2) the character will purchase a new plant. Because neither 1 or 2 have been presented yet in the text, both 1 and 2 are inferences. There is much evidence to suggest that reader's routinely make inferences during reading.

Comprehension processes such as focus and inferencing may be measured *online*, during reading, or *offline*, between or after readings. Online comprehension is characterized by shorter, local reasoning. The reader is attempting to understand just enough of the current situation to grasp the next statement or event. Offline comprehension may contain longer, global reasoning. Here, the reader may reflect upon larger themes or mentally review past events in the narrative. A reader of a novel is engaging in online processing while sitting with the book in his hands, reading sentence by sentence. If the reader sets the book down and begins to analyze the narrative, he engages in offline processing.

1.4 Contributions

This dissertation makes a number of contributions that advance the state-of-the-art in narrative discourse generation:

- *An empirically evaluated formal model of narrative focus:* This dissertation presents a computational model for predicting online narrative focus in plan-based sequences of discourse content. The focus model processes the discourse content by mimicking the reader's comprehension process as described by the Event Indexing model [25, 10] and related models of narrative discourse comprehension [26, 9, 27]. Formal definitions of *foregrounding* - the process by which old information is re-activated - and *updating* - the process by which new information is incorporated - are provided. The model is empirically evaluated in Experiment 1.

- *An empirically evaluated formal model of narrative inferencing:* Also presented is a computational model for predicting online narrative inferences in plan-based sequences of discourse content. The inferences in this model are about the story level, and they consist of causally related sequences of events. The inference model simulates the reader's search after meaning as described by the Constructionist Theory [26]. Two online inference requirements are defined *necessitation* - the inference must be directed towards increasing the coherence of the situation - and *enablement* - the inference must be able to be made easily and quickly. The model is empirically evaluated in Experiment 2.

- *An empirically evaluated method of selecting narrative discourse content to satisfy comprehension criteria:* A generation algorithm that creates narrative discourses to satisfy comprehension criteria concerning narrative focus and inferencing is presented. The generation component defines a partial-order planning algorithm to generate a narrative discourse by selecting and ordering content from an event log. The algorithm defines a method for increasing focus of a particular element and a method for prompting inferences of the type predicted by the MEI model. The results of generation tasks are evaluated in Experiment 3.

- *Implementation of the above models:* The above models are implemented and tested on the experimental data.

1.5 Reader's Guide

The organization of the remainder of this dissertation is as follows. In Chapter 2, the related work in deliberative narrative generation, models of inference in narrative, cognitive models of discourse comprehension, and partial-order planning is summarized.

In Chapter 3, the INFER system is defined. First, the problem statement is given in Section 3.1. The approach and architecture are then presented, followed by definitions of the representations used by INFER. The MEI reader model is presented in Section 3.5. Section 3.5.1 describes the model of focus, and Section 3.5.2 presents inferencing model. Section 3.6 describes the generation component, discussing alternate methods for generation before proceeding to INFER's algorithm in Section 3.6.2. Plan refinement routines, threat detection and resolution, and pruning and heuristics conclude the section and chapter.

Chapter 4 describes the three part experimental evaluation of INFER. Experiment 1 in Section 4.1 evaluates the model of focus. Experiment 2 in Section 4.2 evaluates the model of inferencing, and Experiment 3 in Section 4.3 evaluates sequences of discourse content generated by INFER.

Chapter 5 concludes the dissertation, summarizing the contributions and discussing future work.

Chapter 2

Related Work

There are clearly four sets of efforts related to the work that is described here. Because the goal of this work is to generate narratives to satisfy comprehension criteria, work on the generation of narratives to achive narrative goals is highly relevant. Section 2.1 presents the most relevant efforts in this area. Work on computational inferencing in narrative understanding is presented in Section 2.2. The most related work in cognitive models of narrative discourse comprehension is presented in Section 2.3. Finally, this work employs partial-order planning to construct plans for discourses, and the most relevant areas of this work are presented in 2.4.

2.1 Deliberative Narrative Generation

Narrative generation systems may roughly be divided between *simulation systems*, those that primarily simulate the narrative world, and *deliberative systems* those that primarily deliberate over the choice of narrative elements and events. In simulation systems, the simulation is parameterized by the author and the simulation run is recorded to form the narrative [15]. Recent simulation systems attempt to create realistic world and character dynamics, and many employ complex models of characters [16, 17, 18] [19]. Deliberative systems attempt to construct a narrative that satisfies criteria set forth by the author. Instead of allowing the simulation dynamics to drive the narrative from the bottom up, deliberative systems choose elements and events that satisfy goals and constraints in a top down manner. INFER is a deliberative system; it chooses events to satisfy comprehension criteria. The narrative generation systems reviewed in this section are the most relevant deliberative systems.

Interactive narrative systems allow the user to change the narrative through interaction. Many systems allow the user to play as a character in a story [3, 24] [20], while others employ less direct forms of interaction [28]. Often, these interactive narratives must select from content or create new content in response to user actions, and they use generative components to perform this task. For this reason, some interactive narrative systems are included in this review of narrative generation systems.

2.1.1 UNIVERSE

Lebowitz's UNIVERSE [2, 29] system was one of the earliest deliberative systems. UNIVERSE attempts to generate extended plots for melodramatic fiction using 'plot fragments' and character roles. Figure 2.1 is an example plot fragment about a mean father-in-law or mother-in-law forcing a woman, *?her*, to dump her affair, with *?him*, and remain in a marriage that winds up in divorce anyway. The plot fragment has a list of characters and constraints on those characters. The characters in UNIVERSE's universe are defined to have a list of characteristics in their profile, determining which plot fragments in which they may participate. The plot fragment has a set of goals which it achieves; this fragment churns the relationship between ?him and ?her, preventing them from being happy. The fragment also has a list of subgoals, a partially ordered list of goals which must be met to realize this plot fragment. Though not depicted in this example, fragments may also have a list of updates to characters profiles, expressing how the characters may change during this plot fragment.

UNIVERSE creates plans by hierarchically selecting goals and then choosing plot fragments that achieve those goals. If the initial goal was to introduce disharmony, (churn ?him ?her), then the forced-marriage plot fragment may be selected and instantiated with suitable characters. The subgoals would then be addressed in turn. UNIVERSE attempts to choose plot fragments that satisfy many of the current subgoals in an effort to intertwine plots.

UNIVERSE does not contain a sound and complete planner. Instead UNIVERSE employs a greedy, linear planning method of subgoal decomposition, and, by its author's admission [30], it may plan such that a goal becomes unachievable. Because of this limitation, UNIVERSE works best in open domains with the option of many characters and

```
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) \{prevent them from being happy}
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 (breaking threat)\}
  (do-divorce ?husband ?her)   \{end the unhappy marriage\}
  (or (churn ?him ?her)        \{either keep churning or\}
      (together ?her ?him))    \{tray and get ?her and ?him back together\}
```

Figure 2.1: A UNIVERSE plot fragment ([2] p. 487.)

many plot fragments from which to choose. The resulting narratives contain compositions of plot fragments, and the high control afforded by the plot fragment authoring allows for generated narratives to be interesting and entertaining - though this may place a heavy burden on the author.

2.1.2 DEFACTO

Sgouros's plot manager [20] uses a rule based productions to dynamically create plots with user interaction for the interactive narrative system, DEFACTO. Figure 2.2 is the Plot Manager architecture. The Plot Manager operates in three distinct phases: generation, evaluation, and resolution, attempting to conform to an Aristotealan story arc.

The first phase, generation, uses character role descriptions, the current world state, and rules about the social actions characters may take to create a list of possible actions in the story world. Character roles determine which actions they may take. For instance, the King role may issue decrees to *Forbid* actions, and the Judge role may call upon other characters to be *Punish*ed. Roles are assigned normative goals and actions, such as the Judge wishing to punish characters that have violated a law or decree. Characters

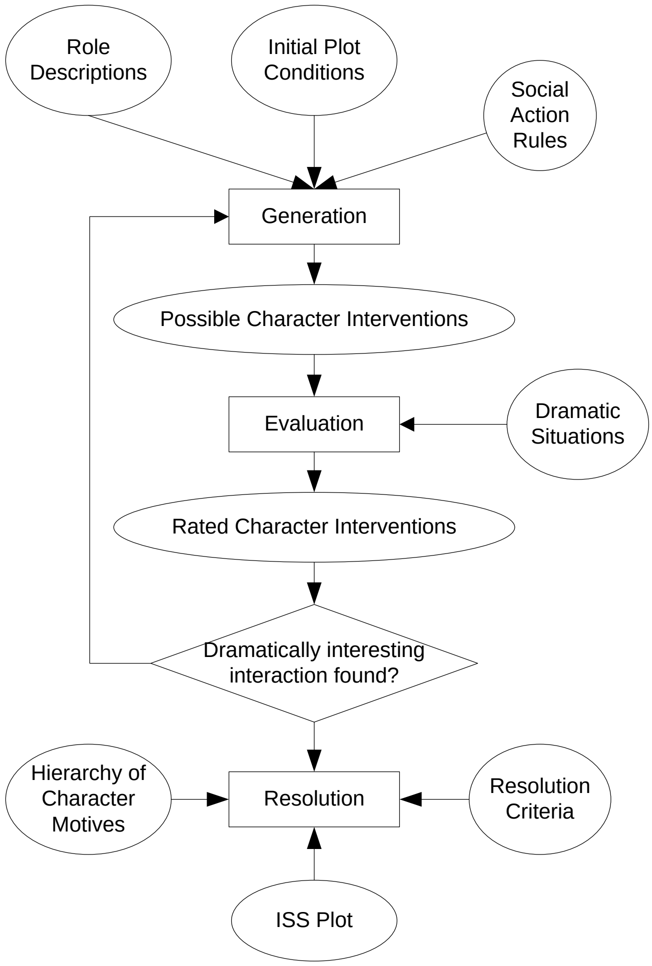

Figure 2.2: The architecture of DEFACTO's Plot Manager.

may also be given individual goals in addition to their roles, such as wishing to worship at an alter. Roles, goals, and norms are defined as first order logic rules. The generation phase allows all characters to generate actions towards their goals in accordance with their role. The collection of all character actions is passed to the evaluation phase to choose which action actually occurs at this moment in the narrative.

The evaluation phase selects a single action from all of the generated actions. The evaluator attempts to select actions that fit into rules defining dramatic situations. For example, if a character has previously been helped by another character, but then becomes opposed by a different character, this constitutes a *Reversal-of-Fortune* dramatic situation. Such a situation would be preferable to other non-dramatic situations. If a possible dramatic situation is discovered, the evaluator chooses that action and returns to generation to create new actions. If no dramatic situation can be discovered, the Plot Manager proceeds to resolve all of the open actions in the resolution phase.

The resolution phase resolves character actions, determining if the actions are successful. Resolution uses a hierarchy of character motives and resolution criteria to attempt to determine the most pleasing resolution to the narrative. Actions associated with motives which are ranked highly are resolved first, and this may cause actions associated with lower ranked motives to fail.

The resulting plots are a product of the simulation aspects of the action generation and the deliberative aspects of evaluation and resolution. Because of the selection of actions to fit dramatic situations, one might expect the resulting plots to seem more dramatic. However, it may be difficult for the author to determine exactly how the interaction of character roles, dramatic situations, and resolution criteria will affect the resulting plot. The dramatic situations presented by Sgouros [20] are simply expressed as combinations of interventions on the part of one character for or against another. These situations are not sufficient to incorporate larger classes of drama, and the interrelatedness of the system does not make the task of defining more nuanced situations approachable.

2.1.3 MINSTREL

Turner's MINSTREL [21] is a narrative generation system that employs a set of transformation routines to model creativity in narrative authoring. MINSTREL uses a schema based representation from Schank and Abelson [31] to represent narrative structure.

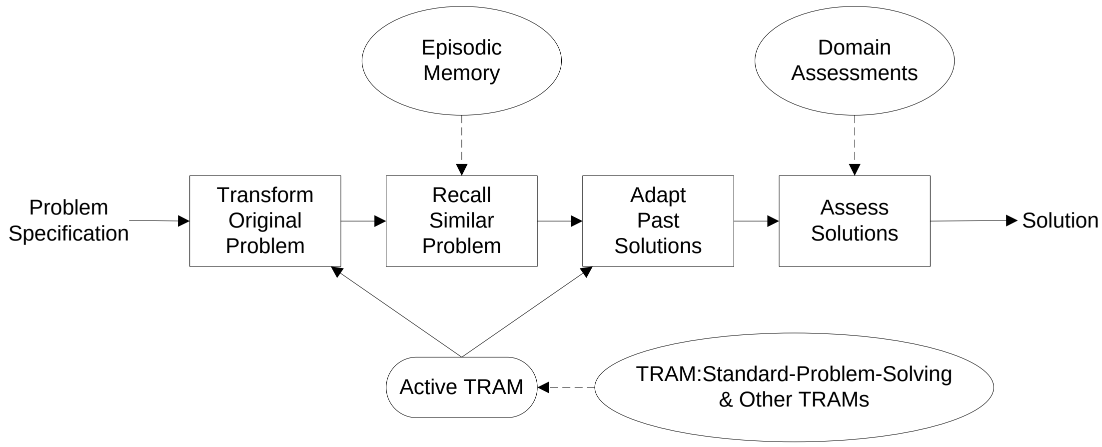

Figure 2.3: MINSTREL's process for generating narratives.

Nodes in the schema graph include actions, character goals, objects, and world states. Links in the schema graph indicate relations of nodes, including Plan, Intends, Precond, Motivates, Achieves, and Thwarts.

Figure 2.3 is MINSTREL's process for generating narratives. MINSTREL uses case-based problem solving to adapt narratives in its database to refine incomplete narratives, employing a library of Transform-Recall-Adapt methods (TRAMs). Each TRAM defines a transformation strategy, transforming the current problem into a slightly modified problem. The database of narratives is searched for instances matching the modified problem, and the resulting instances are adapted for the current context according to the adaptation strategy in the TRAM. For example, the GENERALIZE-CONSTRAINT TRAM removes a constraint from a scene to apply it to a new context. If MINSTREL were given the goal to have a knight commit suicide, and the only narrative in its database contained a knight fighting a monster, resulting in the death of the monster. Then the TRAM would generalize the constraint 'the opponent of the knight must be a monster', recall the fighting scene, and adapt the constraint to 'the opponent of the knight must be a knight'. This TRAM results in the knight fighting and killing himself [21].

MINSTREL also features a set of author level goals for each narrative. These goals drive the creation of the narrative, and MINSTREL applies a set of Author Level Plans (ALPs) to refine the narrative to meet each of these goals. MINSTREL's author goals are categorized into thematic goals, drama goals, consistency goals, and presentation goals. Thematic goals cause the insertion of actions comprising a theme. Drama goals cause the insertion of actions to increase the notion of drama in its ALPs. Consistency goals cause the satisfaction of preconditions of actions in the world. Presentation goals cause the ordering of events for coherent presentation.

MINSTREL presents a novel view on creativity in narrative authoring. The TRAMs and ALPs comprise a set of strategies for creating new narratives by incrementally adapting scenes from old narratives. However, errors may often arise from incorrect adaptations. Turner notes that since a character may hold a book in his hand, MINSTREL may surmise that he may also hold a car in his hand [21]. MINSTREL's collection of TRAMs and ALPs is sizeable, but it do not constitute a principled theory of creativity. Rather, the TRAMs and ALPs present a collection of possible strategies for creativity. Furthermore, the schema representation employed by MINSTREL encodes common sense knowledge appropriate for small narrative domains, but the difficulties of defining consistent schemas may make the authoring of large domains burdensome.

2.1.4 Façade

Mateas and Stern's Façade [3] interactive drama employs a system of small situational plot points, called 'beats', to control the actions and reactions of a pair of characters, Tripp and Grace. The player plays as a guest of Tripp and Grace, visiting their apartment for a casual evening. During course of the evening, the player slowly learns that Tripp and Grace are having marital difficulties as small interactions build tension towards an outcome. The characters' actions are managed by behaviors authored in the ABL behavior language (a reactive planning language based on Hap [32]). ABL behaviors allow for parallel actions to achieve low level goals such as simultaneous walking, talking, and gesturing to change locations and communicate a thought. The goals that the ABL behaviors implement are specified by the current beat.

Beat selection relies on the hand authoring of preconditions and effects over the narrative situation. Figure 2.4 is the set of beat goals (ABL behaviors) for the 'FixDrinks' beat. Each beat goal has preconditions; the TxnIn_AcknowledgeDrinksReferredTo goal may only be activated if the drinks have been referred to. Beat goals may be one of five types:

- transition-in - character expresses desire to pursue an action
- body - prompt the character with a dramatic situation

```
TxnIn_BringUpTheIdeaOfDrinks
TxnIn_AcknowledgeDrinksReferredTo
TxnIn_AcknowledgeDrinkRequested
Body_TripSuggestsAFancyDrink
Body_TripEncouragesRequestedFancyDrink
Body_TripBragsAboutHisFancyDrinks
Body_GraceReactsToTripsBrag
Body_WaitForResponseToTripsSuggestion
Mixin_TripExcitedByPlayersAcceptance
Mixin_TripExcitedByPlayersDifferentFancy
Mixin_TripDismayedByPlayersDecline
Mixin_TripUnsureAboutPlayersReluctance
Mixin_GraceSuggestsCounteringAcceptance
Mixin_GraceSuggestsEncouragingDecline
Mixin_GraceSuggestsCoaxingReluctant
Mixin_TripDiscouragesGracesSuggestion
WaitWithTimeoutForAFinalResponse
TxnOut_TripExcitedPlayerChoseHisFancyDrink
TxnOut_TripExcitedPlayerChoseOtherFancyDrink
TxnOut_GraceExcitedPlayerChoseNonFancyDrink
TxnOut_PlayerChoosesNeitherCompromise
```

Figure 2.4: The set of beat goals for the 'FixDrinks' beat [3] pg. 13.

- mixin - react to a player's actions
- wait-with-timeout - wait for the player to act, or timeout
- transition-out - final reaction to the outcome of the beat

Beats themselves are selected opportunistically by the drama manager. Beats have preconditions and effects that concern narrative history and state in the form of global variables, such as tension. Beats have preconditions restricting the situations in which they may be selected and priority values determining the order in which beats are preferred. The priority values may change according to events in the interaction; beats may become more or less desirable when associated conditions are met. Beats of equal priority are selected according to desired value arcs for tension and character affinity. Façade attempts to increase tension and choose beats that integrate the player's displayed affinity for one of the characters.

Façade has had great success in creating an engaging interactive experience. The beats offer a highly reactive experience while usually managing to convey rising action, climax, and resolution. However, the beat system is designed for minimally structured plot paths, whereas longer, highly structured plots, are difficult to author using beats. Authorship itself is complex and labor intensive [3].

2.1.5 IDTension

Szilas's IDTension system [33, 1] employs a logical formalism to express narrative rules that govern the dynamics of character actions. Figure 2.1 displays three of the rules in the narrative logic. The rules each specify a condition in the world state, and an action that may be carried out. The action may then change the world state upon its execution. For example, in the 'Ending information' rule, the preconditions are: a character, X, must have finished an action, a character, Z, must know X has finished the action, and another character, Y, must not know X has finished the action. In this case, Z can then tell Y that X has finished the action by using the 'inform' action. The 'inform' action results in Y knowing X has finished the action. Much as in the DEFACTO system, rules like these specify all of the possible narrative events, and an evaluation system chooses the best of the possible events.

Name	Preconditions (Left Part)	Triggered Action (Right Part)	Description
Ending information	HAVE FINISHED(X,t,par) KNOW(Z,HAVE FINISHED(X ,t,par)) KNOW(Y,HAVE FINISHED(X,t,par)) different(X,Y) different(X,Z)	inform(Z,Y, HAVE FINISHED(X,t,par))	If a character Z knows that another character X has finished to perform a task, then he could inform a third character Y about this fact.
Encouragement	CAN(X,t,par) KNOW(X,CAN(X,t,par)) KNOW(Y,CAN(X,t,par)) KNOW(Y,WANT(X,t,par)) KNOW(Y,HAVE BEGUN(X,t,par)) KNOW(Y,HAVE FINISHED (X,t,par)) different(X,Y)	encourage(Y,X,t,par)	If a character X is able to perform a task and knows it, if another character Y knows about this possibility but does not know that X has either started or finished the task, then the latter character could encourage the former to perform the task.
Condemnation	KNOW(Y,HAVE FINISHED(X,t,par)) different(X,Y)	condemn(Y,X,t,par)	If a character knows that another character has finished a task, he can condemn him orher for this.

Table 2.1: Example of three narrative rules used in IDtension [1] pg. 769.

The actions of the narrative are evaluated according to their *narrative effects* on the reader. These narrative effects are inspired by readings in narratology, the study of narrative from a language and culture perspective. IDtension calculates a score for the following effects, using a weighted sum to choose the next best action.

- Ethical Consistency - characters which behave in a way showing that they adhere to some ethical values continue to behave according to those values.
- Motivation - characters choose actions which lead to their goals.
- Suitable Complexity - the complexity of the story is measured by the number of possible actions known to the audience. IDtension aims for a moderate amount of complexity.
- Progression - Repetitive actions are penalized.
- Conflict - Internal conflict in characters that are highly motivated to perform a task that violates their ethical code.

The resulting plots are a combination of the rules generating new actions, simulating the world, and the selection of actions for their narrative effects, deliberating over action choice. Because the narrative effects are a linear combination of effect scores, it may become unclear why a particular action is chosen at a particular time. Actions are chosen one by one, partially to accommodate player interaction. The resulting stories may seem to meander as the evaluation score optimizes individual actions, but not sequences of actions. The narrative effects themselves are based upon descriptive narratalogical theories, and may not score effectively for all situations.

2.1.6 Fabulist

Riedl et al.'s Fabulist architecture [34, 4, 35, 36] is a planning based narrative generation system. Riedl et al. have employed Fabulist as an implementation platform for computational theories of narrative generation, three of which are discussed here. First, a model of exploratory creativity expands the search space of possible narratives. Second, vignette-based narrative planning is a case-based reasoning method for generating story plans. Third, the intent-driven partial order causal planner (IPOCL) narrative planner incorporates a theory of character believability into narrative planning.

Firstly, Fabulist includes a model of exploratory creativity to expand the search space of narrative plans [35]. Reminiscent of aspects of MINSTREL's creativity heuristics [21], Fabulist employs two methods for creativity: relaxing constraints on step preconditions and implementing open world planning. Fabulist may remove 'soft' preconditions of steps, such as requiring a actor to be evil before performing an act, to fit characters and objects into roles which they would not normally be accustom. Fabulist also implements open world planning to remove instances where a non-specification in the initial state removes an interesting narrative plan from the search. Instead, in Fabulist, a non-specification in the initial state is taken to mean either the proposition is true or false, and may be decided on later in the planning process. Though MINSTREL's creativity heuristics may be more extensive, Fabulist's techniques generally increase the number of possible narratives from a generation task while maintaining strong causal consistency.

Secondly, Fabulist implements vignette-based narrative planning to model creativity through exploration and retrieval of narrative segments [36]. Fabulist's vignettes are assumed to be minimal examples of narrative sequences taken from another planning domain. The vignettes are recalled to satisfy open preconditions in the planning problem, and they are adapted to the current planning domain using a far transfer algorithm to identify the closest operator. Because the vignettes are considered minimal, every action in the vignette must be included regardless of causal necessity. Though Riedl [36] claims that authors regularly include such causal dead ends in their narratives, perhaps authors have other purposes (e.g. thematic, character identity, emotional) for including actions which appear to have no world state causal consequences.

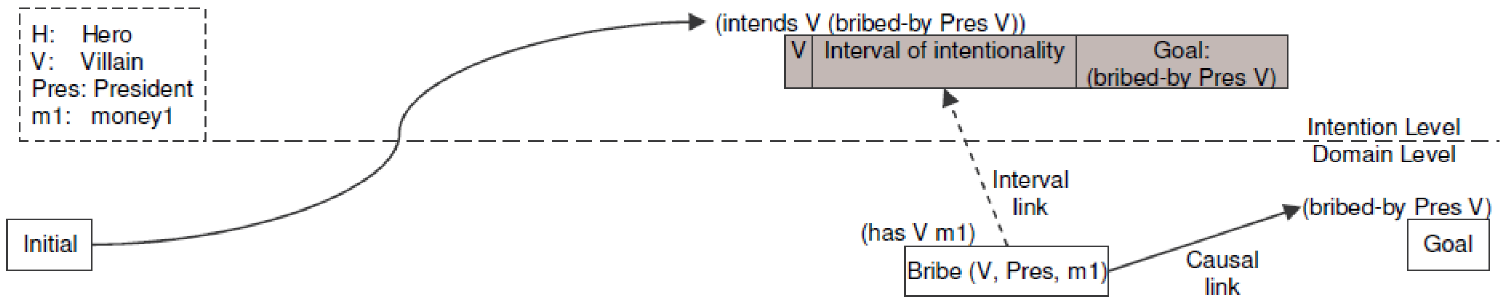

Figure 2.5: A partial IPOCL plan [4].

Thirdly, the IPOCL narrative planner generates story plans that both adhere to a formal model of causality and ensure that characters actions are believable by incorporating a model of character intentions. Figure 2.5 is an example IPOCL plan. IPOCL employs UCPOP-like [8] partial-order planning to create partially ordered sequences of events to comprise the narrative. The events of the narrative are represented as plan steps, which transform a given initial state of the world to a given goal state. Steps contain preconditions, effects, and bindings over free variables. Steps are related by causal links and orderings. Causal links, the bold lines in the figure, denote that an effect of the head step satisfies a precondition of the tail step. Orderings, the dashed lines in the figure, denote which steps come before others.

IPOCL extends partial-order plan formalism to include a *frame of commitment*, an interval in the plan where a character holds a particular intention. Steps may be assigned to frames of commitment via intentional links. Steps assigned to a frame are in service of the frame's goal, and must be on a causal chain leading to that goal. Frames must be motivated by the initial state or steps in the plan; in order for a character to have an intention, it must be caused by some motivating event in the story world.

IPOCL combines a strong representation of causality with a theory of character believability to create more convincing narratives. However, IPOCL does not include a deep representation of character or intention. Characters are defined solely by the intentions they choose, and are not specified by the author. Because of the global knowledge of the planner, characters may take advantage of omniscience. Also, because the plans are constructed in reverse, IPOCL only allows for adopting intentions which eventually succeed. No failed intentions are allowed. Despite these limitations, IPOCL is well suited to reasoning about causality and intentionality in narrative. The work presented here extends IPOCL to model the comprehension processes of readers. More formal definitions are presented in Section 2.4.2.

2.1.7 U-Director

U-Director [5] is a decision theoretic narrative planning architecture. U-Director models narrative objectives, story world state, and user state in a dynamic decision network, used to decide the next action. Figure 2.6 is a slice of U-Director's dynamic decision network. Director actions (which may include null actions) are decision nodes controlled by the system. User actions are chance nodes, probability is assigned to user actions. Both director and user actions update the narrative state, and after each has acted a utility value is assessed. U-Director chooses director actions to produce the highest expected utility after the user acts.

U-Director models Plot Progress and Narrative Flow as part of the Narrative Objectives state. Plot Progress is the progress along the story graph, which actions are ready, waiting, or completed. Narrative Flow models how well multiple actions support a particular goal, and how well multiple actions co-occur in the same location. The Story World state is comprised of Plot Focus and Physical State. Plot Focus models the higher level plot points to which actions are related. Physical State models the user location and the user's activity level. The User State is modeled by the User Goals and Beliefs and the User Experiential State. The User Goals and Beliefs maintains the knowledge to which the user has been exposed, combined with the plot progress and focus. The User Experiential State models the user's independence from director actions (how much the user responds to hints from the director), engagement with the narrative world (how active the user is), and excitement (how much the user's goals and beliefs appear to be changing as well as pacing information). Together, these elements specify a detailed representation of narrative state from which U-Director can decide upon director actions.

U-Director's decision theoretic planning lends itself better to interaction than to pure generation tasks, providing a natural model of uncertainty in interactive narrative. However, its deep representation of narrative state and level of reasoning over that state make it of interest to generation tasks. U-Director balances more traditional plot and story world focused narrative state with an appraisal of user beliefs, goals, and experiential state. This shift towards more detailed models of user experience is continued in later systems, and the concentration on the user experience is central to the work presented in this paper.

2.1.8 Suspenser

Cheong's Suspenser [22, 6] selects discourse content to form a narrative that maximizes feelings of suspense. Suspense level is measured by the number of solutions available to the problems faced by the protagonist. Suspenser uses a partial-ordered plan based representation of a narrative, and a planning based reader model.

Figure 2.7 is the Suspenser Architecture. Suspenser first builds a coherent story

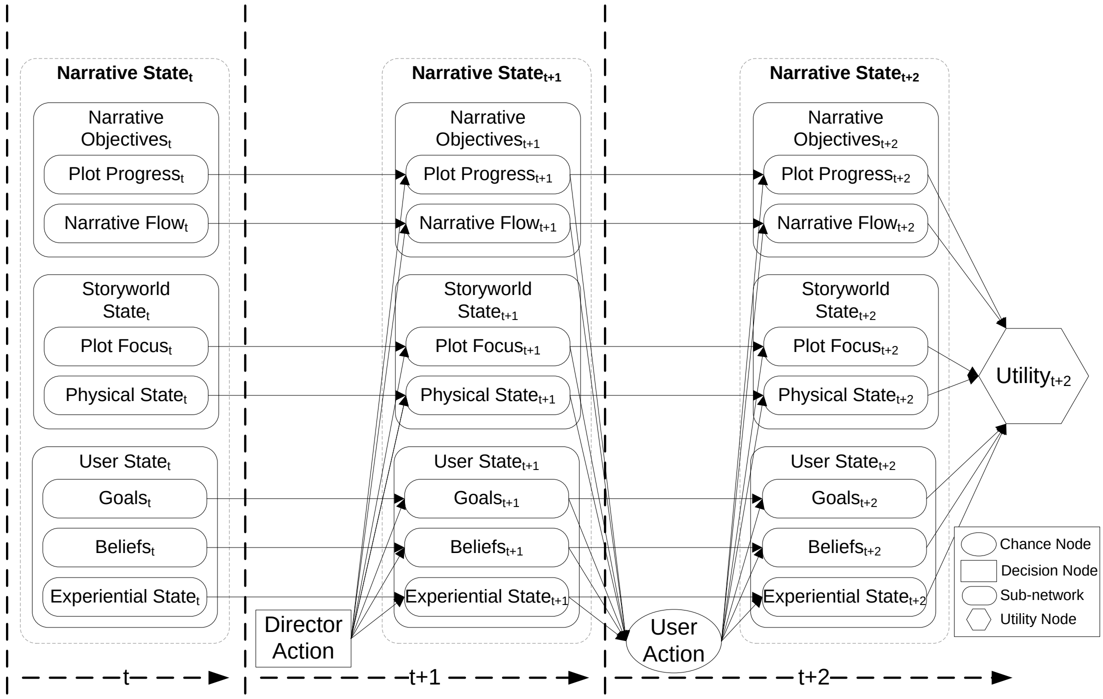

Figure 2.6: A slice of U-Director's Dynamic Decision Network [5] pg 4.

skeleton, a minimal set of events from the narrative to ensure the reader can understand the story. The steps in the story plan are ranked according to their causal relationship to the first event, last event, and goal state. The top N steps form a candidate skeleton, passed to the coherency evaluator. The coherency evaluator uses a partial order planner to model the reader's reasoning process about the story. If beginning with the skeleton the partial order planner can find a complete plan within a resource bound, the skeleton is considered coherent. Otherwise the top ranked step not included in the skeleton is added, and the coherency is checked again. This process repeats until a coherent skeleton is found.

Next, suspense is added to the skeleton by swapping in new actions. The suspense level of a narrative plan is judged by the number of solutions found by the resource bounded planning of the reader model. If the current suspense level is too low, a new suspense-inducing action is found and swapped into the skeleton by replacing the least important event. Suspense-inducing actions are found by a heuristic that counts the number of effects that negate a protagonists goal under the reader's partial knowledge. After the swap, the skeleton is rechecked to see if the new action actually increased the suspense. This process continues until the desired suspense level is reached, or the search space is completely

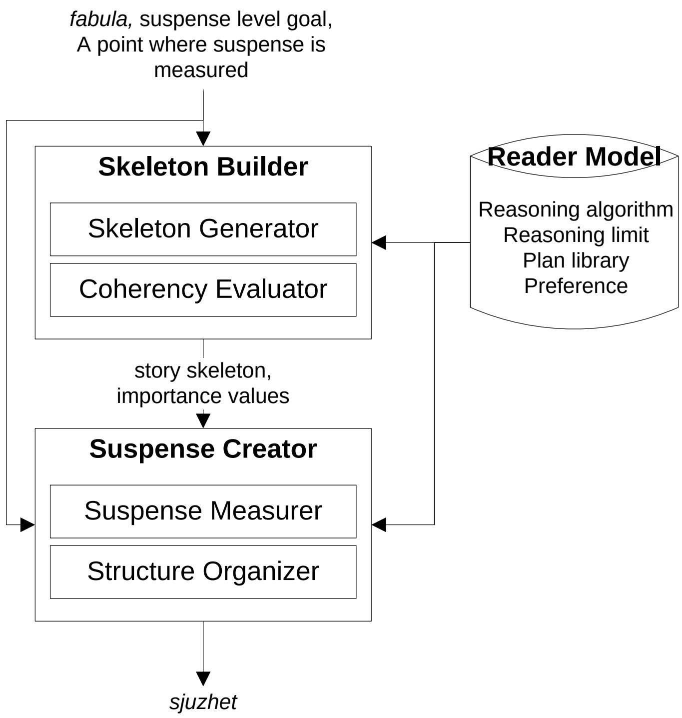

Figure 2.7: The Suspenser architecture [6] pg. 47.

exhausted.

Suspenser's model of the reader's reasoning process for generation purposes borrows from natural language generation techniques [37]. It allows Suspensor to have more direct control over the narrative *comprehension experience* rather than the narrative *structure*, on which other systems have focused. Though Suspenser and INFER approach the reading process differently and with different goals, INFER also incorporates a model of the reader to control the comprehension experience.

2.1.9 Prevoyant

Bae's Prevoyant [23] uses foreshadowing and flashback to create surprise effects in narrative discourses. Surprising events are events that cannot be predicted by the reader, but still make sense in the context of the plot. Like Suspenser [22], Prevoyant employs a partial-order plan based representation of narrative, and a planning based reader model.

To create surprise, Prevoyant first identifies a set of Significant Events (SEs) that may be surprising. The steps with effects causally linked to the goal state are the possible SEs. For each SE, a chain of Initiating Events (IEs) is identified. IE chains are causal chains leading from the initial state to the SE step. Prevoyant attempts to identify IE chains that do not have effects linked elsewhere in the plan, so that if the IE chain is removed, it will not be detected by the reader (no effects will be missing). The SE is then the surprising event. One of the IEs on the chosen IE chain is used to foreshadow the SE, and the entire IE chain is told after the SE providing a flashback. Thus, a surprise event is foreshadowed by a step that leads to the event, then the surprise event is presented, then the entire sequence of events leading to the surprise event is presented in a flashback. Prevoyant checks the unexpectedness of the surprise event using a resource-bounded planner to perform reasoning on the part of the reader.

Prevoyant is similar to Suspenser in its use of a reader model to influence the comprehension experience of the narrative. However, neither Suspensor nor Prevoyant attempt online models of narrative focus and inferencing, and hence their aim and accomplishments are different than those of INFER.

2.2 Computational Models of Inference in Narrative

Many systems have been developed that perform aspects of automatic narrative understanding [38, 39, 7, 40, 41], see [42] and [43] for reviews. However, the INFER system is not directed towards the general problem of narrative understanding, but rather predicting and prompting aspects of narrative focus and inferencing during reading. The systems presented in this section comprise the most relevant set of narrative understanding systems, those that perform some type of story world related inferencing.

2.2.1 SAM and PAM

Cullingford's Script-Applier Mechanism (SAM) [44, 45] uses scripts for everyday action to draw inferences about the events in stories. SAM's scripts are represented in a schema-like fashion as a network of event patterns with causal relations that "describe the major paths and turning points of a common situation" [44]. Scripts contain multiple paths that may lead to an outcome, and the scripts are used to match against information given in a story. When a script is matched against one or more events, intervening steps present in the matched path of the script are inferred. For example, if the story states that "John went to a diner. He left a large tip." (a propositionalized representation is fed into SAM), then the 'restaurant' script would match the initial entering a diner and one of the paths of the restaurant script would match the leaving a large tip. The steps between these two events might include ordering, preparing, serving, eating, enjoying and getting a check. All of these events would be inferred by the script. Another path through the restaurant script may not include enjoying, and might result in a small tip. Multiple overlapping scripts may be active at any one time.

Wilensky's Plan-Applying Mechanism (PAM) [46] extends the script matching concept to deal with character intentions. PAM's scripts are represented in the same schema-like structure, but they are goal centered, describing how a character may achieve his goals. PAM also incorporates 'themes' that motivate goals, such as romantic love motivating the goal to marry. When events occur in the story, PAM applies a rule matching to determine how the events will change the goals of the characters based on their current themes. For example, if the story states that"John is in love with Mary. Mary is kidnapped", then a rule of the love theme may state that John now has the goal of rescuing Mary. Character's actions are interpreted in light of the scripts relating to their intentions, and, as in SAM, steps in the script between events mentioned in the story are inferred.

SAM and PAM apply hard-coded common-sense knowledge to the problem of inferring the relationships between events in the narrative and between events and character intentions. Though these systems are convincing on small examples, scaling to larger, more complex narratives may induce a geometric explosion in the number and variety of scripts that must be authored before inferences can be conducted. INFER employs a planner as a reasoning mechanism, automatically creating sequences from action descriptions. SAM and PAM do not make a distinction between online and offline reasoning. Because of this, these systems make predictions about what the reader may be able to infer, but not what the reader is likely to infer. They would not be well suited for a reader model in online narrative comprehension.

2.2.2 FAUSTUS

Norvig's FAUSTUS [47, 7] employs a general purpose marker passing mechanism to draw inferences about narratives. FAUSTUS's knowledge base is a collection of semantic nets. The semantic nets contain nodes that represent objects, events or or abstractions, and links that relate the nodes via relations such as Dominate, Instance, Argument, Differ. Figure 2.8 is an example of a semantic net used by FAUSTUS.

FAUSTUS processes a narrative one clause at a time, recalling relevant semantic nets for each proposition and fitting the arguments into the nets. Inferences are generated by placing markers in the net, and passing them along the links connecting nodes. Each pass of a marker decreases the marker's energy, until all of the energy is spent and the marker is no longer passed. Markers from separate clauses may collide, indicating that some common inference may be drawn from these causes. These collisions are the inferences predicted by FAUSTUS. For example, if the narrative reads "Bill had a bicycle. John wanted it. He gave it to him.", then the net for having and giving would be recalled for each of the clauses. Marker collisions may indicate the referents of the pronouns (e.g. it refers to bicycle), and the causal and intentional relationships between the actions (e.g. The giving results in John having the bicycle. John wanting the bicycle was the reason for Bill giving it.) [47].

The marker passing of FAUSTUS allows for constructing inferences which are supported by multiple events in the narrative. FAUSTUS's class of inferences is larger than SAM or PAM due to the more general purpose structure of FAUSTUS's semantic nets.

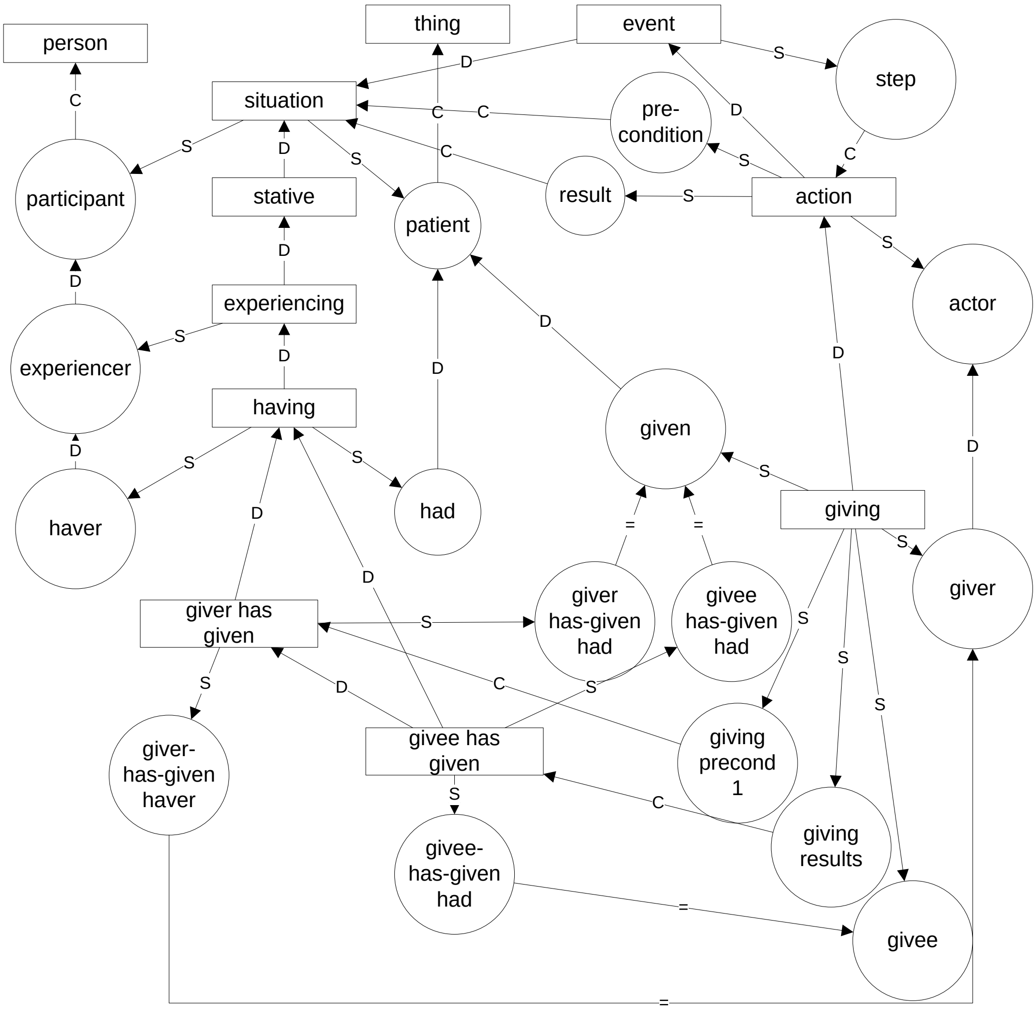

Figure 2.8: A FAUSTUS semantic net [7] pg 71.

However, the static structure of the semantic nets still requires large amounts of knowledge engineering for each new situation. The energy value of the markers used in the inference may some indication of difficulty of constructing the inference, though Norvig does not present any evidence towards this hypothesis.

2.2.3 Distributed Situation Space Model

The Distributed Situation Space model by Frank et al. [41] uses training data about sequences of situations to compute belief values for propositions in related situations. Situations are represented as vectors indicating the truth value of all the propositions in the world. A sequence of example situations is constructed by changing the situation vector according to the effects of events. A world knowledge matrix is created that encodes the probability of a proposition being true in the next situation given the value of a proposition in the previous situation. In this way, the world knowledge matrix contains the probabilistic dynamics of the story world. Inferences can then be computed by determining the likelihood of a proposition between two situations; if the probability is high, then the inference is predicted.

The Distributed Situation Space model is able to reconstruct story world dynamics from a set of example situations, and then use that information to make propositional inferences during comprehension. The limitations of this model include it's reliance on large amounts of training data and the fact that the model has only been tested on microworlds. It does not rely on structured knowledge, and thus would be difficult to implement for generational purposes.

2.2.4 Model-based Story Understanding

Mueller's model-based story understanding [48, 49] employs axioms in event calculus to model the changes in the situations of narratives. Narratives are represented by sets of timestamped fluents that describe the current state in the narrative, and the event calculus axioms determine the events that may change the world state. For example, if the story states that "James places a scarf on the snowman." at time zero, the fluent would be represented as *Happens(PlaceOn(James, JamesScarf, Snowman),0)*. An axiom may state preconditions or effects of an action. So placing the scarf on the snowman may require that James has the scarf, James is awake, James is standing next to the snowman, etc..

The collection of fluents and axioms are used to create a common sense reasoning problem, and a SAT solver is employed to discover all of the possible world models that explain the situation presented by the text. The elements of the discovered models not in the text are the inferences.

Mueller's model-based story understanding uses event calculus axioms and SAT solving to construct all possible world models from the events presented in the text. This technique does not make predictions about the processes employed by readers nor the likelihood of readers making similar inferences about the narrative.

2.3 Cognitive Models of Narrative Discourse Comprehension

2.3.1 Levels of Narrative Representation

Cognitive psychologists have identified multiple layers of discourse structure used in narrative comprehension [9][10]. The separation of discourse structure into levels allows for a separation of the types of reasoning that occur at each level, differentiating the resolution of pronouns from the recognition of narrative themes. Each level uses its own structures to support reasoning, but the levels are not isolated from each other. Inferences at one level may improve understanding at higher or lower levels. Three of the most basic and universal levels are the Surface Code, the Textbase, and the Situation Model.

The surface code is the narrative as presented to the reader. In textual narratives, the surface code is the text itself. Inferences about the surface code might include resolving pronouns or interpreting the meaning of sentences. The textbase is the set of propositions that represent the meaning of the clauses in the surface code. The textbase is the foundation for the situation model, with which INFER is primarily concerned.

The situation model, is a mental representation of the narrative world state; including characters, settings, events in the plot and the diverse interrelationships between these items [9]. van Dijk and Kintsch [50, 10] list reasons situation models are useful for explaining aspects of language processing, a subset of which I include below.

- Models are needed to integrate information across sentences.
- Models are needed to explain similarities in comprehension performance across modalities.

- Models are needed to account for effects of domain expertise on comprehension.
- Situation Models are needed to explain how people learn about a domain from multiple documents. [10]

The situation model is the basis for updating narrative focus and performing inferences about the narrative world. It is formed in the reader's mind during the course of reading, and may consist of elements directly from the text, other knowledge possessed by the reader, and inferences about the narrative. The situation model can be represented as a time related series of individual models, one for each step of processing. Zwaan and Radvansky [10] distinguish (a) the current model of the latest event, (b) the integrated model comprising all of the previous events, and (c) the complete model, which is the final model at the end of the narrative. When a new event is read, a new current model is created to encapsule the new information. The integrated model is then updated to incorporate the current model and to draw new inferences. The specifics of how the current and integrated models are represented and updated differentiate the theories of discourse comprehension.

2.3.2 Narrative Focus

This section examines a selection of the most relevant models of focus in narrative comprehension.

Kintsch developed one of the first general models, the Construction Integration (CI) model [12]. The CI model builds minimally structured associative nets out of the textbase propositions. The CI model, as its name implies, processes new clauses in two phases: construction and integration. In the construction phase, 1) the concepts and propositions directly involved in the linguistic input are formed, 2) the concepts are limitedly elaborated by pulling associated concepts from the general knowledge net, 3) inferring additional propositions, and 4) assigning connection strengths to the lengths in the network (Kintsch does not define the inferencing process). Once the new association net has been constructed, it may be an incoherent or inconsistent collection of propositions and links. The integration phase uses the discourse context to remove inconsistencies and promote coherence. In short, the new association net is overlapped with the previously integrated association net and activation is spread across the network. The nodes that remain activated constitute the most important features of the current discourse representation, they are the subject of the narrative focus.

Langston et al.'s Connectionist Model (CM) [13] operates with principles similar to the CI model. The CM, however, removes the restriction that only working memory nodes are kept for processing and employs a different mechanism for assessing and updating the link weights in the associative net. In the CM model, after the activation has spread across the network, the link weights are updated. Link weights are increased between nodes with high activation, and decreased between nodes with low activation. Thus, activation values depend on link weights, and future link weights depend on activation values. These changes allow for a more dynamic model with to explain experimental results.

Zwaan et al.'s Event Indexing (EI) model [25, 10] specifically for narrative comprehension, separating the possible dimensions of an event into space, time, causality, protagonists and objects, and intentionality. When a new event is read, the event is indexed by the contents of its five dimensions: where the event occurs, when the event occurs, the event's causal relationships, the characters and objects involved in the event, and the character intentions behind the event. Events are then related by the number and types of dimensions they share, and reading new events is easier to the extent which they share dimensions with older events. The assumptions of this model were validated in multiple experiments [25]

Subsequent to the formulization of the EI model an emerging body of research has confirmed, clarified, and extended the EI model. A study by Rinck [51] confirmed that distance and time within the narrative world have a measurable effect on concept accessibility, while text distance did not. Magliano et al. [52] showed evidence that the EI model can accurately predict some aspects of film comprehension. Magliano et al. [53] provided support for the monitoring of character goals and aspects of character focus. Therriault and Raney [54] further explored time in situation models, finding evidence for the representation of durative information. Finally, Mo et al. [55] showed that concepts in a text with more event relatedness but with less mentions were more accessible than concepts with more mentions but less relatedness.

INFER formalizes and implement aspects of the EI model for use in narrative planning, combining the flexibility of the association nets with the dimensionality of event indexing. Events in the discourse are indexed by the five dimensions of the EI model and related in an activation network. From this network, the focus and relatedness of objects and dimensions are calculated. Section 3.5.1 describes this process in detail.

2.3.3 Narrative Inferencing

Graesser, Singer, and Trabasso [26] present a Constructionist Theory of discourse processing; the primary principle of which is that the reader attempts a *search (or effort) after meaning*. When readers comprehend a narrative, they strategically construct representations and reason with these representations to achieve a goal, the most common of which is sense making. This principle is broken into three critical assumptions.

- *The reader goal assumption* states that "the reader constructs a meaning representation that addresses the reader's goals".
- *The coherence assumption* states that "The reader attempts to construct a meaning representation that is coherent at both the local and global levels".
- *The explanation assumption* states that "The reader attempts to explain why actions, events, and states are mentioned in the text". [26]

Graesser, Singer, and Trabasso identify thirteen classes of inferences that may be constructed by the reader [26]. For instance, referential inferences infer that a word or phrase is the same as a previous element, causal antecedent inferences infer previous steps in a causal chain, and thematic inferences infer the main point or moral of the text. The constructionist theory predicts that only a subset of these classes are generated online to satisfy the assumptions above.

The understanding of the production of inferences in narrative comprehension has been confirmed and expanded since the introduction of the Constructionist Theory. Linderholm [56] found that predictive inferences are more likely to be made by readers with high working memory capacity and under conditions of high causal connectivity. Allbritton [57] found that the production of predictive inferences is relatively strategy dependent. Trabasso and Wiley [14] found strong evidence that readers monitor and infer goal plans of actions and make causal inferences based on them during reading. Shears et al. [58] found that character plans are more difficult to construct inferences about than simple causation rules, and Casteel [59] found that predictive inferences are not often generated and not maintain for a long period of time.

INFER embodies many of these theories and extends this work into a formal computational model of inference generation in narrative comprehension. INFER enacts the Constructionist assumptions in the reader's search after meaning, while maintaining the requirements of causal connectedness and conservative prediction for inference production. Section 3.5.2 describes INFER's inference prediction in detail.

2.4 Partial Order Planning

The INFER system uses partial order planning to perform reasoning on the part of the reader, and to generate new Discourse Plans. A brief overview of Partial Order Planning is below, followed by the IPOCL algorithm and representation employed by INFER.

"The task of coming up with a sequence of actions that will achieve a goal is called planning." [60] This work is concerned with classical planning environments that are fully observable, deterministic, finite, static, and discrete. Plans represent states as conjunctions of propositions, goals as partially specified states, and steps (or, actions) that have preconditions and effects over the world state. A planning problem consists of an initial state, a goal state, and a set of possible steps. The planner finds a sequence that begins with the initial state and achieves all of the propositions of the goal state, this sequence is called a complete plan. Classical planning in general the general case is an NP-hard problem [61].

The STRIPS planner [62] is one of the earliest planners, and the STRIPS languge is a basic representation. STRIPS employs the closed world assumption, only allowing positive literals in states; all other literals are considered negative. Step effects include add and delete lists, specifying how the set of true literals is to change. Goals and effects may only be composed of conjunctions of literals.

The ADL planning language [63] removes some of the restrictions of STRIPS, providing more expressibility and power. ADL does not make the closed world assumption. Instead, unspecified literals are treated as unknown. ADL allows for conjunctions and disjunctions in goals as well as existential and universal quantification in goals, and step effects may be conditional. See [60] for examples of planning problems specified in ADL.

There are many types of planning algorithms, each with its own strengths and weaknesses. State space planners search the space of world states via backward or forward chaining. State space planning is easy to implement and runs quickly for small problems. SAT plan algorithms decompose the planning problem into a satisfyability problem and employ a SAT solver. Graph plan algorithms propogate mutex relationships between states and actions to find a solution. The type of planning that most concerns this work, however, is partial order planning.

2.4.1 POP

Paritial-order planners (POPs) are plan space planners that employ a strategy of least commitment. POPs search through the space of candidate plans to find a complete plan. Partial-order plans consist of a set of steps, ordering costraints between the steps, causal links between the steps, and a set of open preconditions. As the name suggests, POPs do not enforce a total order over the steps in the plan, but maintain ordering constraints that may only specify a partial-ordering of the steps. Causal links record causal relationships between steps. A causal link begins at one step's effect and ends at another step's precondition, denoting that the effect satisfies the precondition, and an ordering between the steps is enforced. Causal links may be threatened by interviening steps with effects that undo the link's effect. Open preconditions are those that do not yet have a causal link. A plan in which there are no cycles and no threatened causal links is a consistent plan. A consistent plan with no open preconditions is a complete plan - a solution to the planning problem.

A linearization is a total ordering of steps that satisfies all of the ordering constraints of the plan. If a partial plan is a complete plan, then any linerazation of that plan is also a complete plan.

POP algorithms iteratively refine the plan until all of the open preconditions are eliminated, backtracking when the plan becomes inconsistent. The inital set of open preconditions are the propositions in the goal state. UCPOP is a sound, complete POP for the ADL language [8]. Figure 2.9 is the UCPOP algorithm. To refine plans with open preconditions, UCPOP either reuses a step in the plan or creates a new step from the planning domain with a unifying effect. The a causal link is created between the chosen step and the open precondition. In the case of link threats, UCPOP imposes so that the threat does not occur (promotion or demotion) or separates by ensuring the step bindings can not match. If the plan can no longer be refined, UCPOP backtracks to choose alternate resolution steps or strategies.

Algorithm UCPOP($P =< S, B, O, L >, G, \Lambda$)

1. **Termination**: If G is empty, report success and return P.

2. **Goal selection**: Choose a goal $< c, S >\in G$. If a link $S_i \xrightarrow{e,\neg c} S$ exists in L, fail (an impossible plan). Note that c is universally ground.

3. **Operator selection**: Nondeterministically choose any existing (from S) or new (instantiated from Λ) step S_S with effect e and a universally ground clause $p \in \Upsilon(\theta_e)$ where MGU$(c, p) \neq \perp$. If no such choice exists then fail. Otherwise, let $L' = L \cup \{S_S \xrightarrow{e,c} S\}, B' = B\cup$ MGU$(c,p) \cup B_e \cup B_S$, $O' = O \cup \{S_S < S\}$, $G' = G- < c, S >$, and let $S' = S \cup \{S_S\}$.

4. **Subgoal generation**: If effect e has not already been used to establish a link in L with bindings MGU(c, p) then let $G' = G$ and for each $\theta \in \Upsilon(p_e\backslash$ MGU$(c, p))$ add θ, S_S to G'. If $S_S \notin S$ for each $\theta \in \Upsilon(p_S\backslash$ MGU$(c, p))$ also add θ, S_S to G'.

5. **Causal link protection:** Let l bea a causal link $S_i \xrightarrow{e_i,q} S_j$ in L. Let S_k be any step with an effect e_k and postcondition $p \in \Theta_{ek}$. Step S_k THREATENS likn l with clause $v_p \in \Upsilon(p)$ if possibly $S_i < S_k < S_j$ when MGU$(\neg q, v_p) \neq \perp$ is consistent with B. For all such S_k, e_k, l and v_p such that S_k threatens l with v_p, nondeterministically do one of the following (or, if no choice exists, fail):

   (a) **Promotion** If possibly $S_j < S_k$, let $O' = O \cup \{S_j < S_k\}$.

   (b) **Demotion** If possibly $S_k < S_i$, let $O' = O \cup \{S_k < S_i\}$.

   (c) **Separation** Let $O' = O \cup \{S_i < S_k < S_j\}$ then nondeterministically

      i. Choose constraints β' on existentially quantified variables such that MGU$(\neg q, v_p) = \perp$ and let $B' = B' \cup \beta'$, or

      ii. Choose a precondition $\tau \in \Upsilon(p_{Ek}\backslash$ MGU$(\neg q, v_p))$ and let $G' = G' \cup \{< \neg\tau, S_k >\}$.

   (d) **Recursive invocation:** If B' is inconsistent then fail; else call UCPOP$(< S', B', O', L' >, G', \Lambda)$.

Figure 2.9: The UCPOP algorithm [8].

2.4.2 IPOCL

The INFER system employs the IPOCL planner and IPOCL plan representation as part of it's reasoning process. IPOCL and its plan representation are defined below.

The intent-driven partial order causal link (IPOCL) planner [4] generates narrative plans to satisfy plot coherence and believability. For IPOCL, plot coherence is the property of having a consistent plan, and character believability is the perception that character actions are motivated by the character's intentions. IPOCL extends the POP representation to include the frame of commiment, representing a character's intention to achieve a goal. Frames must be motivated by steps or propositions in the intial state, so that characters have a reason for their intentions. Steps in the plan performed by characters must be part of a frame of commitment. Steps which are part of a frame are in service of the frame's goal, on a causal path that leads to the fulfillment of that goal. Figure 2.5 is a partial IPOCL plan with two frames of commitment and three steps.

Figure 2.10 is IPOCL's algorithm for generating narrative plans. IPOCL augments the UCPOP algorithm in Figure 2.9 with an extra routine in causal planning and adding motivational and intent planning routines. In causal planning, steps are selected or added as before, and new frames of commitment are created for the step, and new intent flaws are created to add the step to a frame. In intent planning, steps are assigned to frames, and in motivation planning, frames are motivated by preceeding steps. The algorithm returns when there are no open preconditions, all steps are part of at least one frame, and all frames are motivated.

The frame of commitment, IPOCL's representation of intentionality, separates IPOCL plans from UCPOP's [8] partial order plans.

Definition, Interval of Intentionality *is a tuple,* $< S, c, g_c, s_f >$ *such that* $s_f \in S$*, where* S *is a set of plan steps,* c *is a single character (denoted by unique name),* g_c *is an internal character goal held by* c*, and* s_f *- referred to as the final step of the interval - has* g_c *for an effect and does not temporally precede any other step in* S*.*

Definition, Frame of Commitment *is a tuple,* $< c, g_c, I >$*, where* c *is a single character (denoted by unique name),* g_c *is a single internal character goal that* c *is committed to, and* I *is an interval of intentionality which shares the same character and internal character goal with the frame of commitment. [4]*

An interval of intentionality is a set of steps that a character performs in service

Algorithm IPOCL($< S, B, O, L, C >, F, \Lambda$)

1. **Termination.** If O or B is inconsistent, fail. If F is empty and $\forall s \in S, \exists c \in C | s$ is part of c, return $< S, B, O, L, C >$. Otherwise, if F is empty, fail.

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

   - **Causal Planning**
     - **Goal Selection.** Select an open condition flaw $f =< s_{need}, p >$ from F. Let $F' = F - f$.
     - **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_{add} 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 $L' = L \cup \{< s_{add}, e, p, s_{need} >\}$. If $s_{add} \neq s_{old}$, add new open condition floaws to F' for every precondition of s_add.
     - **Frame discovery.** Let $C' = C$.
       * 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, let s_{add} be part of c, let $C' = C' \cup \{c\}$, create a new open motivation flaw, $f =< c >$, and let $F' = F' \cup \{f\}$.
       * Let C'' be the set of existing frames of commitment that can be used to explain s_{add}. For all $d \in C''$, creat an intent flaw $f =< s_{add}, d >$ and let $F' = F' \cup \{f\}$.
     - **Threat resolution.** Performed in the standard way.
     - **Recursive invocation.** Call IPOCL($< S', B', O', L', C' >, F', \Lambda$)

Figure 2.10: The IPOCL Algorithm, part 1 [4].

- **Motivational Planning**
  - **Goal Selection.** Select an open motivation flaw $f =, c >$. Let p be the condition of c. Let $F' = F - f$.
  - **Operator selection.** Same as causal planning above, except $O' = O \cup \{s_{add} < s_i | s_i \text{ is part of } c\}$.
  - **Frame discovery.** Same as for causal planning.
  - **Tqhreat resolution.** Performed in the standard way.
  - **Recursive invocation.** Call IPOCL($< S', B', O', L', C' >, F', \Lambda$)
- **Intent Planning**
  - **Goal selection.** Select an intent flaw $f =< s, c >$ from F. Let $F' = F - f$.
  - **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 each $s_{pred} \in S$ such that $< s_{pred}, p, q, s > \in L$, create an intent flaw, $f =< s_{pred}, c >$. Let $F' = F' \cup \{f\}$.
    * Do not make s part of c.
  - **Recursive invocation.** Call IPOCL($< S', B', O', L', C' >, F', \Lambda$)

Figure 2.11: The IPOCL Algorithm, part 2 [4].

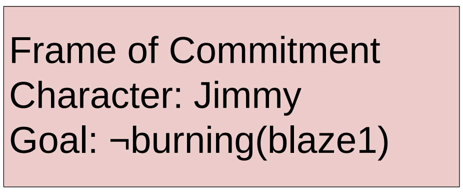

Figure 2.12: An IPOCL Frame of Commitment.

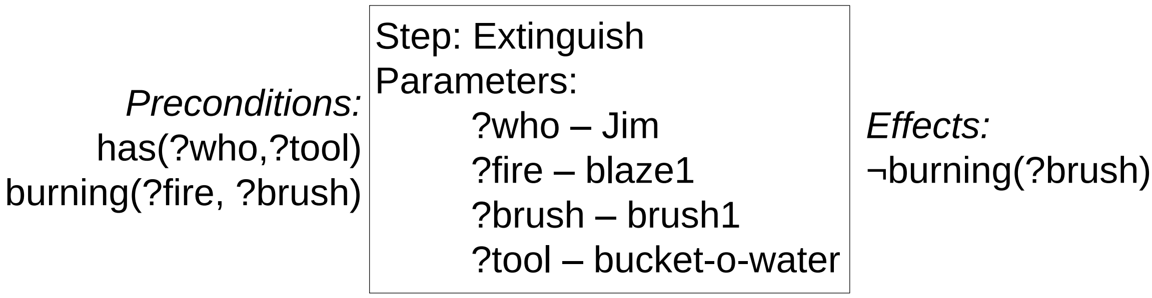

Figure 2.13: An IPOCL Step.

of the goal, g_c [4]. The interval of intentionality ensures that the frame is motivated, the intention is established in the plan, and it allows steps to be assigned to the interval through intention links.

Definition, Effect *is a triple* $< p, \theta, \beta >$ *wher* p*, the effect preconditions, is a set of quantified literals;* θ*, the effect postconditions, is a set of quantified literals; and* β*, the effect binding constraints, is a set of equality constraints on free variables in* p *and* θ*. [8]*

Definition, IPOCL Step *is a tuple* $< p, e, c, B >$*; where* p *the step preconditions, is a set of quantifed literals;* e*, the step effects, is a set of effects;* c*, the step actors, is a set of variables denoting the actors for the step; and,* B *the step binding constraints, is a set of co-designation and non-codesignation constraints on the free variables in* p *and* e*. [4]*

An IPOCL Step is similar to a UCPOP step, except that it includes a list of actors for the step, those characters which are said to be intentionally participating in the step. IPOCL will attempt to assign the step to frames of commitment for all the characters in c. c may be the empty set, in which case the step requires no assignment.

Definition, Ordering Constraint *is a pair* $< s_i, s_j >$*, denoted* $s_i < s_j$*, where* s_i *is ordered before* s_j*. [8]*

Definition, Causal Link *is a quadruple* $< s_i, e, r, s_j >$*, denoted* $s_i \xrightarrow{e,r} s_j$*, where* r *is a precondition of* s_j*, and* e *is an effect of* s_i*, and* $\exists q \in \beta_e$ *such that* q *unifes with* r*.*

Definition, IPOCL Plan *is a tuple,* $< S, B, O, L, C >$*, where* S *is a set of steps,* B *is a set of binding constraints on the free variables in* S*,* O *is the set of ordering constraints on steps in* S*,* L *is a set of causal links between steps in* S*, and* C *is a set of frames of commitment.* [4]

IPOCL plans contain steps with binding constraints, causal links, and ordering constraints as well as frames of commitment with intervals of intentionality. Together this structure defines a partially-ordered set of events that are related to a simple representation of character intentionality.

Definition, IPOCL Plan Completeness *An IPOCL plan is complete when O and B are consistent, every step is part of a frame* ($\forall s \in S, \exists c \in C | s$ *is part of* c)*, every frame is motivated, and there are no open preconditions.*

A plan is complete if and only if (1) all preconditions are all steps are established, (2) all causal threats are resolved, and (3) all steps are intentional.

Definition, IPOCL Planning Domain *is a set of IPOCL steps.*

A step s in the planning domain can be instantiated into the plan by creating a copying s_c, adding binding constraints to $s_c(B)$ and setting $S = S \bigcup s_c$. The event library is used to model narrative inferencing by giving the system a set of types of events that might be inferred.

Chapter 3

The INFER System: An Approach to Generating Discourse Plans that Satisfy Comprehension Criteria

3.1 Problem: Defining a Model of Narrative Comprehension

The means by which readers understand the story world and the events of the narrative are of primary interest when generating a narrative discourse. In order to use models of narrative understanding for generation, some critical questions must be addressed. What aspects of the story world do readers most often represent, and how is this information collected and assimilated? What properties make narrative discourses hard or easy to understand, and what properties engage the reader in the narrative? This section identifies fundamental modes of representing and reasoning about the discourse and story world that may contribute to constructing computational models of narrative comprehension. This section concludes with an analysis of the state of the art in narrative understanding and generation, identifying gaps between current work and the goals of narrative generation for comprehension.

The Situation Model is the reader's representation for the current narrative situation . The Situation Model should be represented to perform the types of processing that readers commonly perform during reading, including focus and inferencing. Narrative focus is a measure of the salience of a narrative element in the readers mind. The salience of an element is partially determined by the recency of reading and the relevancy to the current situation. Narrative inferencing is the process by which readers add information to the narrative. One prominent form of inferences are events or sequences of events that may happen in the story world but are not present in the discourse at the current point of reading. Inferences must be necessitated, required to understand the context, and enabled, easily deduced, to be reliably constructed during reading. Inferences may be necessitated by causal gaps in the discourses sequence of events; these are termed causal inferences. Inferences may also be necessitated by character intentions, these are termed intentional inferences. These forms of processing the discourse are essential to generating discourses to fulfill desired comprehension criteria.

Readers learn about the story world one fact or event at a time. A reader processes a single clause using the current context and then integrates the new information with his beliefs about the story world. The amalgamation of integrated information that results is called the Situation Model [10]. The Situation Model is the basis for reasoning about the story; readers use the Situation Model both to shift narrative focus and construct story world inferences [10] [26]. A reader model for narrative understanding and generation should represent the Situation Model in such a way as to allow for the common types of reasoning that readers perform. In online reading comprehension, the current Situation Model should be constructed after each element is read, as the reader does, and common forms of processing should be computed at that time.

Narrative focus, also known as foregrounding or activation, is a measure of the salience of a narrative element in the readers mind. Readers only possess a limited amount of attention, and they cannot keep every character, object, fact, and event in the forefront of their mind [11]. Instead, the elements which the reader finds most important at any time are kept in focus and less important elements drift further from the center of attention. The elements that are in focus are most readily accessible to the reader and often form the basis for comprehension and the construction of inferencing [11] [13] [10].

The salience of a narrative element has been shown to relate to a few factors. Recency of reading is a strong indicator of activation. If the element was the last thing the reader read about, the reader can often access it easily. Relatedness to the current context is also a strong indicator of activation. If a character picks up a paint brush, walks across an art studio, has a conversation, and then approaches an easel, the paint brush will most likely be activated in the reader's mind. In this situation, the paint brush is related to the

1. Jimmy played guitar in a band.

2. Jimmy's guitar was stolen.

3. He wanted to record a new hit single.

4. He saved all of his money in the bank.

5. He drove to the guitar shop.

6. He drove to the studio.

7. He tuned his new guitar.

Figure 3.1: An example of a narrative that may prompt inferences.

character (he is carrying it), the character's presumable intentions (he wants to paint), and thematically related to the other items in the environment (easels and paint brushes are often found together). The effect of relatedness on activation has been noted often in the literature, for many different categories of relations [12] [13] [25] [51] [52] [53] [54] [55].

A computational reader model may include narrative focus as a means to compute ease of comprehension and likely inferences. Such a reader model should represent recency and relations between narrative elements, which is non-trivial. Activation networks have been employed for this purpose with some success [12] [11] [13].

There is much evidence to suggest that reader's routinely construct inferences during reading. Inferences add information not present in the discourse to the Situation Model in order to construct a more complete view of the narrative situation. Reader's may use inferences to resolve the referents of noun phrases and pronouns, character goals, determine themes, fill in gaps in causal chains [26], or predict the actions of characters [14]. Consider the example in Figure 3.1. In this narrative, the reader may infer that Jimmy purchased a guitar at the shop, and that Jimmy will proceed to record a song. Without inferencing, this narrative may seem to be a random collection of facts and events. However, by constructing these inferences, the narrative presents a more cohesive whole, engaging the reader in the problem solving task of understanding.

Inferences about the story can be constructed by assessing the current situation, identifying gaps in the knowledge about the story world, and attempting to fill in the gaps with the most plausible information. This process is often performed by the reader in the course of understanding the narrative with little or no conscious effort. Inferences may be *elaborative* or *necessitated*. Elaborative inferences are inferences which increase the knowledge of the story world, but are not needed for comprehension. Readers do not routinely make elaborative inferences [26] [27]. Elaborative inferences may enhance the experience of the narrative, however, they are not necessary to understand the narrative and may often prove to be untrue. Necessitated inferences are those that aid in fully understanding the narrative. Such inferences are constructed online by readers more frequently and reliably [26] [27].

The Constructivist Theory uses three assumptions to predict inferences [26]. *The reader goal assumption* states that "the reader constructs a meaning representation that addresses the reader's goals." *The coherence assumption* states that "The reader attempts to construct a meaning representation that is coherent at both the local and global levels". *The explanation assumption* states that "The reader attempts to explain why actions, events, and states are mentioned in the text." [26] These assumptions define further criteria for necessitated inferences. The reader attempting to understand the narrative to fulfill his own goals, make sense of the story as it unfolds, and explain the intent behind the inclusion of elements.

Causality and Intentionality are central concepts to the understanding of the story world. Causality is directly related to the coherence assumption. If the reader cannot identify a continuous causal chain of events, they may be unable to place events in space and time or understand their relationship with each other. To maintain coherence, readers can construct causal inferences by inferring events when causality breaks down. Intentionality is directly related to the reader goal assumption and the explanation assumption. The intentions of the characters directly impact on their actions. A reader may form the goal of predicting the outcome of the story, a common phenomena of so-called 'page turner' novels. The reader's success in this goal is largely related to his ability to predict the actions and reactions of characters, and thus understand their intents. Also, readers may seek to explain why a character performs a certain action. For both of these reasons, readers may construct inferences based on character intentions to determine the future (in the discourse) actions of characters and justify present actions.

A reader model may want to incorporate inferencing to gauge the coherence and cohesiveness of story elements or to better understand the reader's mental state during and after reading. To incorporate online inference prediction, a reader model may define criteria for necessitation and enablement by embodying assumptions of the Constructivist Theory [26] or similar theories. The concepts of causality and intentionality should be strongly considered when predicting inferences, due to their relationship to coherence, reader goals, and the explanation of elements.

A narrative representation to support such a reader model must make considerations for reader model processing. The representation should have separate structures for story and discourse to support the linear processing of the discourse and the reconstruction of the story. The story should have a basic representation of objects, characters, facts, and events to model the story world. These elements determine the situations that may be reconstructed in the Situation Model. To support inferencing with causality and intentionality, the events should have causal and temporal relationships between each other as well as intentional relationships to the goals of characters. The story representation should present a complete picture of the relevant facts and events in the story world. The discourse, however, may present a partial view of the story.

The reader model described in this section processes the discourse elements one by one and makes online predictions about the reader's focus and story world inferencing. The reader model builds an activation network from the sequence of elements to predict focus. Inferencing is predicted by reconstructing the story from the elements of the discourse, and attempting to identify when inferences are enabled and necessitated. The causal structure of the story is examined to determined gaps in causality to prompt inferences. The intentional structure of the story is examined to determine opportunities for the characters to act upon their intentions, prompting inferences.

Current cognitive models of narrative comprehension are not sufficient for the purpose of generation. Such, cognitive models are human created frameworks that tie together theory and observations about narrative understanding [26] [10] [11] [64] [27] [13]. They rely upon researchers' efforts in authoring narratives, creating metadata describing aspects of those narratives, and then testing theories using the narratives and data in experiments. Few aspects of these models are described in sufficient rigor for implementation on a computer, and those that are do not constitute a complete, end to end processing framework. Though some of these models define algorithms for narrative focus [11] [13], none of them define algorithms for narrative inferencing, and thus no systems exist that incorporate both focus and inferencing. The models often describe the workings of average case scenarios, which are most often seen in the data, but do not take time to deal with the edge cases so essential to algorithmic behavior. Thus, new theory is required to bridge the gap between cognitive models of narrative comprehension and computational models of narrative comprehension for the purposes of generation.

No current narrative generation systems employ cognitive models of online narrative comprehension. However, many plan-based generation systems incorporate formalisms that may be of use in such a system. The Mimesis interactive narrative system [24] used the Longbow partial-order planner to generate narratives. The STRIPS-like partial-order plans contained causal links, representing causal relationships and orderings representing temporal relationships. The IPOCL planner for the Fabulist architecture [34] [4], extended the Longbow planner to include character intentions and a theory of character believability. IPOCL plans represent both causal relationships and character intentions. The planner has the ability to construct causal chains of events, but IPOCL does not include a model of the reading process to explain narrative focus or when and where inferences may be constructed.

Other notable generation systems incorporate implicit or explicit reader models, though, again, none are cognitive models of online narrative comprehension. Mott's U-Director [5] employed a decision theoretic planner to choose the next best event according to criteria of narrative progression through a plot graph, story world state, and the system's model of the user's state. The user's state is defined by a model of the user's goals, beliefs, and experiential state, and more detailed models of the users have been explored in subsequent research [65] [66] [67]. This user modeling is distinct from reader modeling in that it attempts to understand the actions of the user in a virtual world, not the reader's narrative comprehension. Cheong and Young's Suspenser [22] generated narrative discourses from a plot representation to fulfill a model of suspense. In Suspenser, suspense was rated by the inverse of the number of solutions available to the problems of the protagonist. If the protagonist had few or no solutions to his problems, suspense was rated as high, if he had many solutions to his problems, suspense was rated as low. Bae and Young's Prevoyant [23] use flashback and foreshadowing to reader surprise. Surprise is measured by a computation of unexpectedness, whether the reader can detect missing events, and postdictability, whether the story makes sense afterwards. Though both Suspenser and Prevoyant model some aspects of narrative comprehension for dramatic effect, neither define a model of narrative focus or the criteria for prompting inferences.

Defining a computational model of online narrative comprehension is a significant problem that has yet to be solved. Cognitive models of narrative comprehension are not defined in sufficient rigor for computational comprehension algorithms. Narrative generation systems have developed formalisms for causality and intentionality and have employed reader models, but none have addressed the narrative focus and inferencing in online narrative comprehension, both of which are fundamental to narrative understanding.

3.2 Approach

The proposed solution, the INFER system (INferences For Extending Recall), generates narrative discourses to satisfy comprehension criteria concerning narrative focus and inferencing. Cognitive models of narrative comprehension are not directly applicable for use in generative systems due to lack of rigor. INFER incorporates theory from cognitive models and extends this theory to form a computational model of online narrative comprehension. Past narrative generation systems have developed formalisms for causality and intentionality and have employed reader models, but none have addressed narrative focus and online inferencing. INFER simulates the reading process by constructing a Situation Model, calculating narrative focus and predicting inferences in the form of causal chains of story world events. INFER then uses this reader model to generate narrative discourses.

3.3 Architecture

INFER generates narrative discourses to satisfy comprehension criteria concerning narrative focus and inferencing. INFER models narrative focus by spreading activation from recently mentioned elements to related elements. Relations between elements are represented in a semantic net. This formalization is inspired by the Event Indexing (EI) model [25], a cognitive model of narrative comprehension. INFER models some aspects of narrative inferencing by generating sequences of events to either bridge gaps in causality or bridge gaps between the current world state and character intentions. This inferencing follows the principle that readers constantly search for connection and meaning in the narratives that they read, a principle that is central to the Constructivist Theory [26]. INFER generates a sequence of discourse content from a collection of facts, objects, and

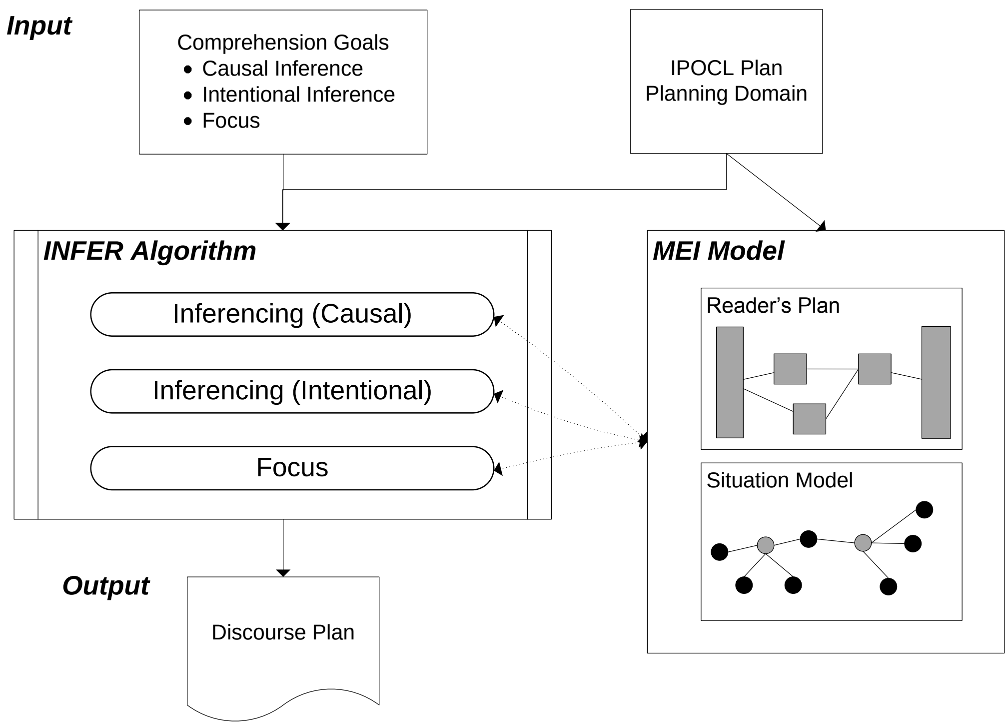

Figure 3.2: The INFER architecture.

events, attempting to achieve desired comprehension effects, either by shifting focus to a particular event or character or prompting desirable inferences.

Figure 3.2 depicts the architecture of INFER. INFER constructs a discourse level representation with desired comprehension properties from a story level representation. INFER searches the space of possible discourses to find one that satisfies all of the comprehension criteria. The inputs are an IPOCL plan, a planning domain, and a list of comprehension criteria. INFER consults the MEI reader model to make predictions for the reader's comprehension of the current discourse, and to choose elements that will prompt the desired comprehension effects. For each comprehension criterion, INFER uses the MEI model to determine which sets of elements might be included to achieve the desired effect, changing focus or prompting an inference. INFER chooses one of these sets of elements for inclusion in the discourse and proceeds with the next comprehension criteria. If at any point the MEI model cannot find a satisfying set of elements then the search must backtrack. The resulting sequence of discourse content is a totally ordered subset of objects, characters, intentions, and events from the story plan.

The inputs to INFER together specify a satisfying set of discourses. The narrative events, objects, characters, facts about the world, and intentions that eventually constitute the output discourse are drawn from the IPOCL plan. The planning domain is a set of action templates that define which events are possible in the narrative world. These templates cover the possible inferred events for this domain, and are used predict the construction of inferences. The comprehension criteria define the intended causal inferences, intentional inferences, and focus requirements of the discourse. Once the reader model predicts the criteria are satisfied, the discourse is complete.

The IPOCL plan is the story structure from which every element in the discourse is taken. An IPOCL plan is like a UCPOP partial-order plan in that it contains a propositional representation of initial and goal states, steps instantiated from an operator library, bindings over each step, orderings, and causal links [8]. What makes the IPOCL plan unique is its assignment of intentionality for each action taken by characters. This intentionality is necessary for the modeling of intentional inferencing - reasoning about the future actions of characters based on their intentions.

The planning domain is the set of events possible in the narrative world. It is used to make the predictions to model how the reader may infer events and actions not directly represented in the discourse.

The comprehension criteria define the intended effects of the discourse on the reader, and allow the problem author to choose which elements of the narrative will appear most prominently in the discourse. They are of three types: intentional inference criteria, causal inference criteria, and focus criteria. The intentional and causal inference criteria specify exactly which inferences are to be prompted by the discourse. The focus criteria specify which elements are to be most salient in the reader's mind during and after reading. A discourse that satisfies these criteria will be predicted by the reader model to prompt all of the inferences in the inference criteria and maintain a minimum amount of focus on the elements in the focus criteria.

INFER uses the MEI reader model to predict the effects of the discourse on the mental state of the reader. The MEI model consists of the *Situation Model*, a semantic network to compute focus, and the *Reader's Story* an IPOCL plan to maintain the reader's understanding of the story. The situation model and Reader's Story are constructed incrementally as each element from the discourse is processed. The situation model maintains the connections between elements along the dimensions of space, time, causality, protagonists and objects, and intentions. It uses these connections to model how focus spreads from one element to the next during comprehension. The Reader's Story represents the reader's view of the story structure as the discourse unfolds. The Reader's Story is used as the starting point for generating inferences after each element of the discourse is processed.

Thus the reader model makes three types of predictions, one for each type of comprehension criteria. These are: which inferences are prompted by the causal structure of the story, which inferences are prompted from the intentional structure of the story, and which elements are most salient in the reader's mind. These predictions are used to answer three types of questions during generation: which elements might be included to prompt for a causal inference, which elements might be included to prompt for an intentional inference, and which elements might be included to draw focus to a particular element.

INFER generates discourses by searching through the space of possible discourses, solving one comprehension criterion at a time. If the search reaches a criterion that cannot be satisfied by adding to the current discourse, it must backtrack to consider other options. First INFER attempts the causal inference criteria, asking the MEI model which elements to include to prompt causal inferences. Next it attempts the intentional inference criteria, asking the MEI model which elements to include to prompt intentional inferences. Thirdly, it manipulates focus. INFER queries the MEI model to address each of the elements in the focus criteria.

Section 3.4 defines the data structures used by INFER. Section 3.5 discusses the MEI reader model. Section 3.5.1 describes the model of focus. Section 3.5.2 presents inferencing model. Section 3.5.3 describes the construction of the Reader's Story. Section 3.5.5 describes the model of inferences prompted by the causal structure of the story, and section 3.5.6 describes the model of inferences prompted by the intentional structure of the story. Section 3.6 discusses generation. Section 3.6.1 discusses alternate methods for generation. Section 3.6.2 describes INFER's method of generation. Section 3.6.3 describes the refinement routines used for satisfying comprehension criteria. Section 3.6.4 describes threat detection and resolution. Section 3.6.5 describes pruning and heuristics.

3.4 Definitions

3.4.1 INFER Input

The input to INFER is an *Event Log*, an *Event Library*, and a set of comprehension criteria. The event log is defined as an IPOCL plan, the library of possible events is an IPOCL planning domain, and the comprehension criteria is divided into focus, causal inferencing, and intentional inferencing criteria.

The *Event Log* is represented by a complete IPOCL plan [68]. Thus, the event log is not only a listing of all the events that have occurred in the story world. It is comprised of the events, objects, facts, and characters as well as causal, ordering, and intentional relations between them. The *Event Library* is an IPOCL Planning Domain, a set of templates for creating new steps in the plan.

The comprehension criteria defines when a discourse is predicted to create the desired comprehension effects in the reader. Focus criteria describe which elements to highlight in the narrative. Inference criteria describe which inferences are desirable in the narrative. Causal and intentional inference criteria have the same form, but are treated separately because the reasoning processes used to satisfy them are assumed to be different.

Definition, Focus Criterion *A focus criterion is a tuple* $< e, x >$ *where* e *is a proposition, inference, or event and* x *is a number. The focus criteria means* e *is to have a total activation of at least* x*.*

The focus criteria define minimum activation for elements in the narrative. An element that is identified in a focus criterion is guaranteed inclusion and a specific amount of predicted attention by the reader. By inclusion in a focus criterion, an element can be guaranteed more centrality to the discourse according to the reader model.

Definition, Causal Inference Criterion *A causal inference criterion is a tuple* $< s, b >$*, where* s *is an IPOCL step and* b *is a set of bindings over the free variables in the step.*

The causal inference criteria define which steps are to be inferred by the reader model (and hence the reader) due to the causal structure of the narrative. As defined below, in causal inferencing readers attempt to fill in noticable gaps in the causal structure of the narrative. A step that is indicated in a causal inference criterion is predicted to be inferred by the reader and may or may not appear in the event log. The bindings over the free variables given in the criterion may range from no bindings to completely bound.

Definition, Intentional Inference Criterion *An intentional inference criterion is a tuple* $< s, b >$*, where* s *is an IPOCL step and* b *is a set of bindings over that step.*

The intentional inference criteria define steps to be inferred by the reader model (and hence the reader) due to the intentional structure of the narrative. As defined below in intentional inferencing, readers attempt to predict character actions based on stated intentions. A step that is indicated in a intentional inference criterion is predicted to be inferred through the process of intentional inferencing and may or may not appear in the event log. The bindings over the free variables given in the criterion may range from no bindings to completely bound.

These criteria allow for the specification of predicted aspects of narrative comprehension, but they are predicated on the ability to predict activation, causal inferences, and intentional inferences. The criteria are only satisfied in reference to a model of the reader. The MEI model, a computational model of narrative comprehension, is defined in Section 3.5 to perform these computations.

3.4.2 INFER Output

The output of INFER is a sequence of discourse content, a total ordering over a subset of the elements in the event log. The sequence of discourse content represents a proposition level telling of a narrative which may be directly translated to a medium such as text or film. The sequence may be constructed from an Event Log input by choosing a subset of the elements from the Event Log and then choosing a total order over these elements.

Definition, Discourse Plan *is a tuple,* $< S, B, O, L, C, D, I >$*, where* S *is a set of steps,* B *is a set of binding constraints on the free variables in* S*,* O *is the set of ordering constraints on steps in* S*,* L *is a set of causal links between steps in* S*,* C *is a set of frames of commitment,* D *is the subset of* S *which are causally prompted inferences, and* I *is the subset of* S *which are intentionally prompted inferences.*

Definition (Sequence of Discourse Content) *A sequence of discourse content is a tuple,* $< R, S, B, O, L, C, T >$*, where* S *is a set of steps,* R *is a set of propositions from the preconditions and effects of the steps in* S*,* B *is a set of bindings on the free variables in* S*,* O *is a set of ordering constraints on steps in* S*,* L *is a set of causal links between* *steps in S, C is a set of intentional frames of commitment, and T is a total ordering over $S \bigcup R \bigcup C$.*

Definition (Narrative Element) *A narrative element is a plan step, proposition, or frame of commitment.*

A Discourse Plan represents a plan for generating a discourse. Discourse Plans are structured as IPOCL plans with denotations for inferred steps which are to be removed from the final discourse. INFER uses Discourse Plans as an intermediate representation for generation. Before output, INFER chooses a linearization of the Discourse Plan (minus inferred steps) to serve as the Sequence of Discourse Content that is the final output. Any linearization of a Discourse Plan is a satisfying Sequence of Discourse Content.

The Sequence of Discourse Content contains three types of elements: steps, propositions, and frames of commitment. The steps represent events in the narrative such as “Jim kicked the ball”. The propositions represent single facts about the world such as “The vase rested on the table” and may also serve to introduce and identify objects in the world: “There was a girl named Sally”. The frames of commitment identify intentions of the characters as in “Bob wanted to eat some ice cream.”. The total ordering over these elements asserts the order in which they should be presented to the reader. Thus, if a ordering over step s proposition p and frame of commitment f specifies $s < p < f$, then the step s is presented, followed by the proposition p, followed by the frame of commitment f. Figure 3.3 shows a visualization of an example sequence of discourse content.

The sequence of discourse content in figure 3.3 can be used to create a narrative in a medium such as text or film by realizing each discourse element within the medium. A simple template based system might be used to translate each element into a sentence for a text, or a visual discourse generator may select a sequence of shots and actions to convey the discourse. Figure 3.4 shows the translation from figure 3.3 to a text, performed by hand.

3.4.3 Reader Model Definitions

The reader model employs a weighted semantic network referred to as the *Situation Model* to represent the focal aspects of comprehension, defined as follows.

Definition (Event Node) *An event node is a vertex.*

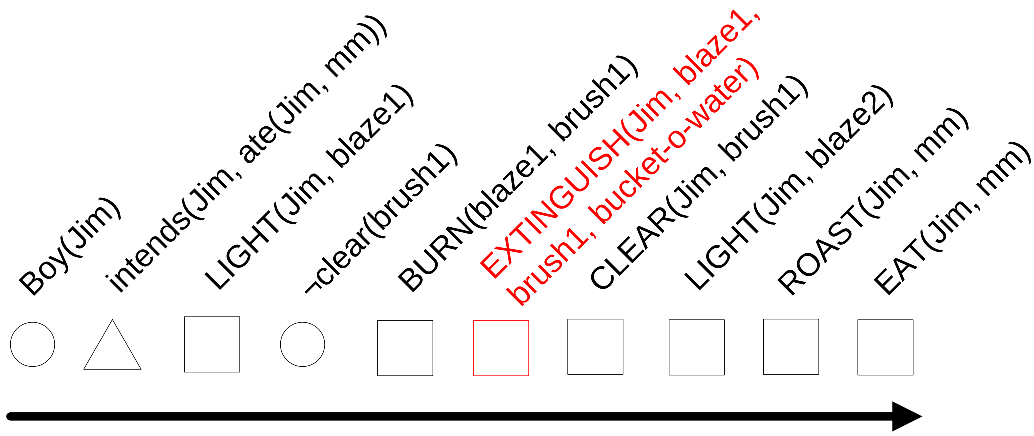

Figure 3.3: A Sequence of Discourse Content. Events are squares, intentions are triangles, and propositions are circles. The sequence, read left to right, expresses how Jim attempts to roast marshmallows (mm) but causes a small brush fire instead. Jim then extinguishes the brush fire, clears the brush, and roasts and eats the marshmallows.

1. There was a boy named Jim.
2. Jim wanted to eat a marshmallow.
3. He lit a fire,
4. but the surrounding brush was not properly cleared!
5. The brush caught fire.
6. Jim quickly put out the fire.
7. Then he cleared the brush out of the way.
8. He relit the fire,
9. and roasted his marshmallow.
10. He ate his marshmallow.

Figure 3.4: The sequence of discourse content from Figure 3.3 realized in text. Realization completed by hand.

Definition (Dimension Nodes) *There are five types of dimension nodes: space, time, causal, object, and intention. Space, time, and causal nodes contain no extra information. Object nodes contain a proposition. Intention nodes contain a character and a proposition.*

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

The Situation Model is defined as a bipartite graph linking events nodes to dimension nodes. The dimension nodes represent relevant aspects of the events: space, time, causal relatedness, characters and objects, and character intentions. So if the character Mary takes part in five events in the story, then the character node representing Mary will be linked to five event nodes in the situation model. Section 3.5.1 describes how the Situation Model is constructed and how it is used to compute focus.

Definition (Reader Story) *A Reader Story is an IPOCL plan.*

The Reader's Story is the reader's representation of the events of the story world. Hence it is represented the same as the event log. Since the reader often does not know the whole story, however, it need not be a complete plan. Section 3.5.3 describes how the Reader's Plan is constructed and how it is used as a starting point for inferencing.

3.5 MEI, The Reader Model

The Modified Event Indexing (MEI) model represents the reader's processing framework and memory during the reading process. The MEI model is a computational model of online comprehension motivated by findings in cognitive models of narrative comprehension. The two parts of the MEI model are 1) the *Situation Model*, a semantic network of nodes and edges to maintain the reader's shifting focus as new events are read, and 2) the *Reader's Story* an IPOCL narrative plan to maintain the reader's understanding of the story world as the discourse is read. The situation model and the Reader's Story include events, orderings, causal links, the characters and their intentions, and the objects and locations in the world.

The MEI model predicts aspects of online comprehension, which is distinct from offline comprehension. In online comprehension, the reader is constantly engaged in understanding the narrative, reading and comprehending simultaneously [11] [10]. The reader works to understand singular events and facts as well as the surface level relationships between them, but does not stop to consider deeper meanings or propose future situations. Thus, the reader pauses only seldomly and for short periods of time in online comprehension. In contrast, readers may also engage in offline comprehension of narratives. In offline comprehension, the reader pauses the reading of the discourse and turns his full attention to understanding the broader implications of what has been read [11] [10]. The reader may consider many aspects of the narrative such as theme, character intentions, motivations, or the author's intent, and the reader may think on the narrative for a long period of time.

In online comprehension, readers use recency of reading and the relatedness of narrative elements to aid in fast comprehension. Narrative elements that were recently mentioned are generally read faster and more easily comprehended, as are elements that are highly related to the current situation [11]. Faster reading and easier comprehension are two measures of the salience of an element in the reader's mind, and these effects are useful to a computational model of narrative comprehension because of their strong predictions over the reader's mental state.

Inferencing in online comprehension is both limited and targeted. Readers do not generally make unconstrained predictive inferences during reading because they often prove to be untrue and they are usually not necessary to understand the narrative situation. Hence, these elaborative inferences do not often advance the reader's goal of comprehending the narrative, and in this sense, they are wasted effort. Instead, readers may make shorter bridging inferences to understand the purpose of elements in the narrative or to comprehend the context.

In following these aspects of online comprehension, the MEI model models the recency and relevancy of narrative elements to compute salience, and it predicts inferences about the story world which are both limited and targeted at the reader's comprehension goals. The following sections describe the MEI model in detail.

3.5.1 Focus

The MEI processing of the discourse content mimics the reader's comprehension process as described by the Event Indexing model [25] [10] and related models of narrative discourse comprehension [26] [9] [27]. The MEI model provides a formal definition of *foregrounding* - the process by which old information is re-activated - and *updating* - the process by which new information is incorporated.

The Situation Model is built incrementally from the steps in a sequence of discourse content. It is divided into the integrated model, the events and dimensions already processed, and the new model, the most recent event to occur. After each element is read, the new model representing that element is created and the integrated model is *updated* to include the new model. Updating occurs in discrete time, once after each element.

The following principles from cognitive models of discourse comprehension guide the the formal model of focus.

1. The recency and relatedness of elements determines their salience. Focus can be modeled by an interconnected graph of concepts with weighted edges. This represents elements from the narrative and connections between them. This has been a common approach in discourse processing [25] [12] [13].

2. The strength of the connection between two events is a function of the number of shared dimensions of those events. This is the major claim of the Event Indexing model [25].

3. After reading each element, focus spreads from the new element to related elements in an amount proportional to the strength of connection between them [12] [13].

4. Thus, elements that are continuously related to new events retain or improve focus over time, and elements which are not continuously related to new events lose focus over time [12].

The new model is created according to the type of the last element read, l_i. If the element is a proposition, a dummy event node is created and object nodes for the proposition and the arguments are created and attached to the event node. Similarly, if the element is an intention, a dummy event node is created and intention nodes for the intention and object nodes for each argument are created and attached to the event node.

In the case where the element is a plan step. The new model, N_i, is obtained directly from the most recent plan step, l_i, as follows: Create a new event node e_i. Create a new space node d_i^s, time node d_i^t, and causal node d_i^c. For each proposition p_j in the

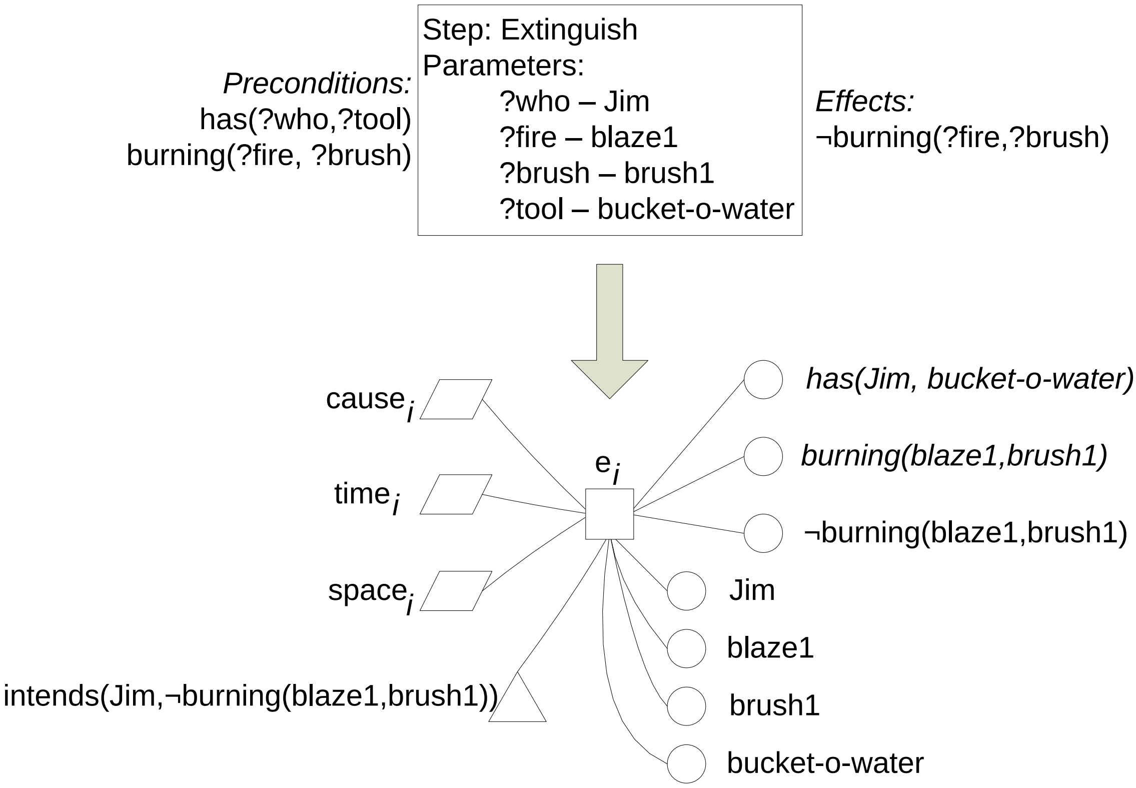

Figure 3.5: A transition from a plan step to a new model.

preconditions and effects of s_i, create a new object node $d^o_{i,j}$, and for each term t_k in p_j create an object node $d^o_{i,j,k}$. Create an intention node d^n_i for each of the intentions to which s_i is linked.

For example, Figure 3.5 shows the transition from the plan step representing "Jimmy puts out the blaze with a bucket of water" to the new situation model for this event. Dimensional nodes are formed for the causality of this event, the spatial location of this event, the temporal location of this event, and Jimmy's intentions during this event. The causal, time, and space nodes are place holders to link an event to other events that have a causal relationship, occur close in time, or occur close in space. Dimensional nodes are also created for each of the bindings - the Jimmy protagonist, the blaze1 object, the bucket-o-water object - and for each of the propositions in the preconditions and effects. Finally, a new event node e_i is created and linked to each of the dimensional nodes (link weights are not depicted), forming a star graph. The new links are all given the weight of

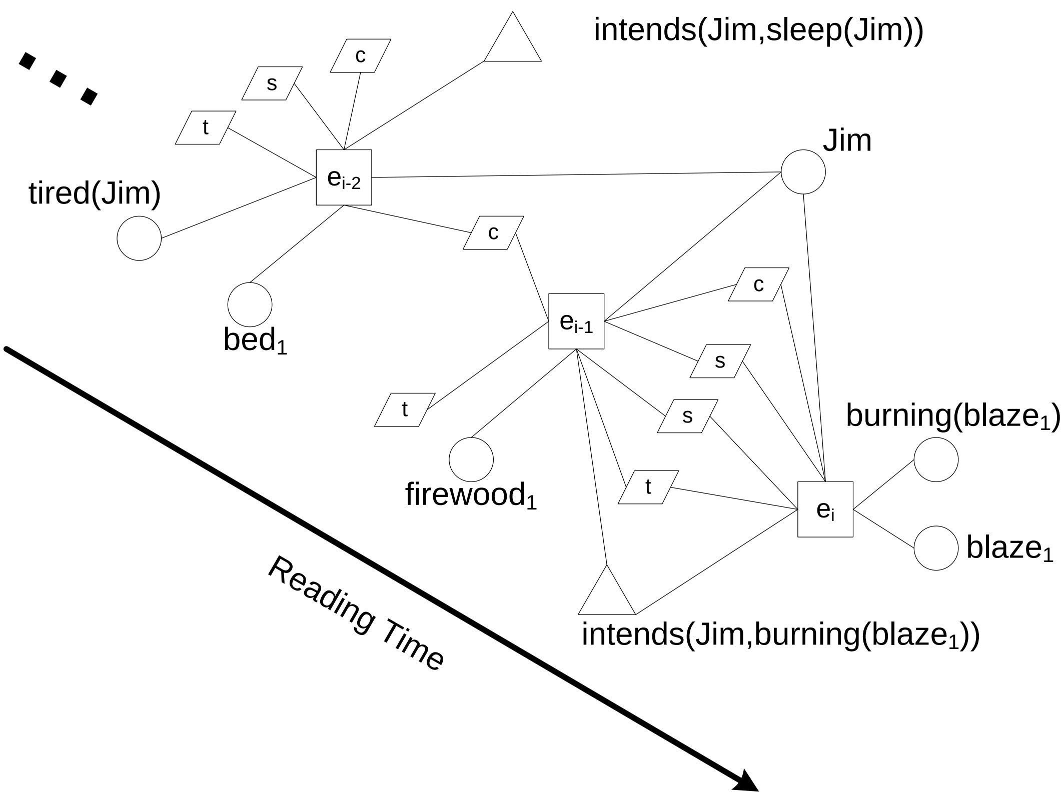

Figure 3.6: A Situation Model during processing. Events nodes are squares, intention nodes are triangles, proposition nodes are circles, and time, space, and causality nodes are rhombuses.

1.0.

Once the new model has been created, the integrated model is then updated by incorporating the new model and simulating to obtain shifts in focus. The initial integrated model is empty of nodes, and it is expanded over the course of reading by incorporating each new model as it is created. Figure 3.6 shows an example integrated situation model; the last three events and associated dimensional nodes are depicted. Incorporating the new model and simulating focus is described immediately below.

Let the integrated model at time i be G_i and the new model at time i be N_i, the update function, $\biguplus$, is defined as:

$$G_{i+1} = G_i \biguplus N_i =< V^G_{i+1} =< X^G_{i+1}, Y^G_{i+1} >, E^G_{i+1}, w^G_{i+1}(e) > \tag{3.1}$$

All of the nodes and edges are combined into a single graph:

$$\begin{aligned} X_{i+1}^G &= X_i^G \cup X_i^N \\ Y_{i+1}^G &= Y_i^G \cup Y_i^N \\ E_{i+1}^G &= E_i^G \cup E_i^N \end{aligned} \tag{3.2}$$

Object nodes are considered to be the same node if their propositions are the same or are codesignated. Update the edge weights of G_{i+1} by 1) discounting the existing weights and 2) increasing the weights of the edges closest to the new event node, e_i.

$$w_{i+1}^G(e) = \begin{cases} w_i^N(e) & e \in E_i^N \\ S(D(w_i^G(e)), e, e_i) & \text{otherwise} \end{cases} \tag{3.3}$$

D is a discrete approximation of an exponential discounting function. Letting d be a constant:

$$D(x) = x * d \tag{3.4}$$

S increases the weights of the edges closest to e_i. It uses the shortest-path-distance function SPD to determine the distance between the edge e and the event node e_i. A simple step function F discounts the effects as the edges are further away: Let c_f be a constant:

$$F(x) = \begin{cases} 1/x & x < c_f \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

S can now be defined in terms of these functions. Let v be a node adjacent to e along a shortest path to e_i with minimal degree, $D(v)$ be the degree of v, and c_s be a constant, then the spreading weight function as is defined as:

$$S(x, e, e_i) = \frac{F(SPD(e, e_i))}{D(v)} * c_s + x \tag{3.6}$$

By which the weights on the edges closest to the new event node are increased the most, and then less as the edges become further. The increase is divided equally among vertices connected to the same node.

The discounting of older edges and the spreading of weights from the newer edges simulates the backgrounding of older information and the foregrounding of newer information in the reader's mind. The weights on edges that are far from new nodes in the graph erode with each time step. In contrast, edges that are close to new information in the graph are constantly refreshed.

Thus, the MEI model incorporates the new situation models derived from plan steps into the integrated models. The graphs representing each are joined, and the link weights are updated as weight spreads from the new event node and deteriorates on nodes that are no longer referenced. The situation model graph computes the activation value for each node in the network, as predicted by the EI model [10]. The activation is the prediction of an element's salience in the reader's mind. Nodes that are more activated are more likely to be recalled and are more important for the interpretation of new events. Activated nodes are predicted to have faster access times in word recognition tests.

The activation of any node in the network can be computed by finding the value of this node in the Markovian steady state distribution, π. The situation model can be represented as a continuous-time Markov process by numbering each of the nodes and constructing a transition rate matrix, Q. This treatment follows that of Stewart (1994) [69]. Entry $q_{ij}, i \neq j$ in Q is equal to the weight of the edge between node i and node j in the semantic model, or 0 if there is no edge. This is made a conservative process by setting

$$q_{ii} = -\sum_{j \neq i} q_{ij}. \tag{3.7}$$

Also note that this is a stable process, since all the edge weights are defined to be finite.

To find the steady state distribution of Q, and thus the activation value of each node, I first find the Embedded Markov Chain (EMC), S of Q. S is the discrete-time Markov chain with the same steady state distribution of Q. Each entry s_{ij} of S is defined to be

$$s_{ij} = \frac{q_{ij}}{\sum_{j \neq i} q_{ij}}. \tag{3.8}$$

S can thus be written as

$$S = I - D_Q^{-1} Q, \tag{3.9}$$

where $D_Q^{-1} = diag\{Q\}$, the diagonal matrix of Q.

The stationary probability vector, ϕ (a row matrix), can be found by solving

$$\phi(I - S) = 0, \tag{3.10}$$

for ϕ such that the 1-norm of ϕ, $\| \phi \|_1 = 1$. The 0 on the right side of this equation is a row matrix of 0's. From ϕ and D_Q^{-1}, I can find the steady state distribution, π, as

$$\pi = \frac{-\phi D_Q^{-1}}{\| \phi D_Q^{-1} \|_1}, \tag{3.11}$$

The π is then normalized into a unit vector to obtain the relative activation value of each of the nodes. It can be found for the instantaneous situation model after each time step, and its calculation ensures that the most heavily linked nodes are the most activated for any one time point.

These calculations can be approximated by assigning each *node* in the graph an initial activation value of 1.0, and then iteratively spreading the node weights according to the weights of the links between the nodes until the maximum delta drops below an error threshold.

To compute the ease of reading a new fact or event, the activations of the nodes to which it will be linked are averaged. This results in a *relatedness score*. This method is used to predict reading times in the experiments.

3.5.2 Inferencing

Inferences predicted by the MEI model are causally related sets of events. They are represented as sets of IPOCL steps with causal links, ordering constraints, and binding constraints. To create inferences about the narrative world from the perspective of the reader, the events, characters, and intentions are represented as a partial IPOCL plan referred to as the *Reader's Story*. The reader's story contains all of the steps, propositions, and frames of commitment in the sequence of discourse content up to the current point of reading. For each new element processed, a copy of the reader's story is made, and causal and intentional inferencing, defined below, are employed to identify the new sets of steps that constitute the inferences.

The MEI model's inferencing algorithms follow a small set of guiding principles that have been observed in psychological studies of discourse processing.

1. Inferences in online comprehension must be necessitated and enabled. Readers do not generally make unconstrained predictive inferences during reading because 1) they often prove to be untrue, 2) they are usually not necessary to understand the narrative situation, and 3) may involve long, time-consuming reasoning. Hence, these elaborative inferences do not often advance the reader's goal of comprehending the narrative, and in this sense, they are wasted effort. Instead, readers may make shorter bridging inferences to understand the purpose of elements in the narrative or to comprehend the context [70].

2. Readers more often make inferences about elements that are in focus. Graesser et al. (2002) state "Inferences are more likely to be made if they are more recent events on a causal chain (i.e. they more directly lead to an event being explained) and if the inferences are connected to many other events on the chain." Both the recency and relatedness of elements increase focus in the MEI model.

3. Readers may make inferences that are incorrect. Predictions about unknown properties of a narrative may prove false, even when the possibilities are highly constrained.

4. Inferences aid in comprehension [26]. Comprehension often requires the readerto fill in the gaps presented in the narrative, and inferences allow the reader to do so.

3.5.3 Constructing the Reader's Story

Incrementally constructing a Reader's Story P_r from a discourse plan with the order preserving constraint is relatively straight forward. Each element, e_i, in the sequence of discourse content, T_d, is processed in order:

- If e_i is a step, add e_i and all of its bindings and causal links to its preconditions to P_r. Add an ordering from each step and frame of commitment proceeding e_i to e_i (as per the order preserving constraint).

- If e_i is a character intention, then create a new frame of commitment f_i in P_r. Add an ordering from each step and frame proceeding f_i to f_i (as per the order preserving constraint).

- If e_i is a proposition, create a dummy step s_i with the sole precondition and sole effect of e_i. Add s_i to P_r as above.

After each element has been added to the Reader's Story, the causal and intentional inferencing algorithms are run to determine the inferences which are currently prompted by the Reader's Story.

3.5.4 Simulating Online Inferences

This section defines the criteria for determining when two types of inferences, causal and intentional, are necessitated and enabled. Inferences consist of single steps or sequences of steps that will be predicted or assumed to have happened by the reader. Causal inferences are steps inferred to maintain causality. Causal inferences are necessitated when the world state changes without explanation. The reader can infer that something must have happened to change the world state. Intentional inferences are steps inferred to fulfill a character's stated intention. Intentional inferences are necessitated when characters have unfulfilled intentions. The reader can infer that a character will act to achieve his goals. Both types of inferences are enabled when the preconditions of all the steps in the inference can be fulfilled by the elements in the discourse, when the search for the inference is small and well constrained, and when the elements involved in the inference are in focus.

Determining when to construct inferences as well as what information to construct inferences about is a significant problem. Inferences could be constructed at any time during reading, and they may be about anything in the story world. Consider the discourse in Figure 3.8. The reader may choose to construct inferences concerning the color of the front door, the number of keys Billy had, the reasons for Billy entering the door, the reasons the front door was locked, or that Billy unlocked the front door using his keys.

Myers et al. suggest that unprompted inferences are more often made when inferencing is *necessitated* to understand the story and *enabled*, so that the inference is not difficult to construct [71] [10]. Graesser and Clark [70] note that readers rarely make unconstrained predictive inferences and that when they are asked to, the inferences often prove to be untrue. For the restricted view of online inferencing in this work, causal and intentional inferences are constructed when they are both necessitated and enabled.

Causal inferences are necessitated when the truth value of propositions changes without an intervening step. Consider Figures 3.7 and 3.8. In Figure 3.7, the reader does not know whether the door was locked at the beginning of the story or not. The truth value of LOCKED(FRONTDOOR) is asserted to be false by the opening of the door, but it has not changed from true to false. A causal inference is not necessitated to understand the discourse. In Figure 3.8, LOCKED(FRONTDOOR) is asserted to be true as the first line in the discourse, but then the character enters the front door. Entering the door requires the door be unlocked, and the truth value of LOCKED(FRONTDOOR) *is* changed from true to false. At this point a causal inference is necessitated to determine how the door became unlocked. The reader might infer that Billy unlocked the door using the keys.

Intentional inferences are necessitated when characters have unfulfilled intentions. In Figure 3.9, the reader learns of Adams intention to get something to eat. At this point,

1. Billy grabbed his keys.
2. Billy entered the front door.

Figure 3.7: Example narrative without causal necessity.

1. The front door was locked.
2. Billy grabbed his keys.
3. Billy entered the front door.

Figure 3.8: Example narrative with causal necessity.

the intentional inference of how Adam will fulfill his intention of getting something to eat is necessitated (though perhaps not enabled).

Both causal and intentional inferences are enabled when the reader has enough information to create the inference and the reasoning process is short enough to be quickly accomplished by the reader. In the inferencing presented in this work, the truth value of propositions in the story is never assumed or inferred. Instead, sequences of events which transition the story world between known states are the only form of inferences. In order for these inferences to be made, the discourse must have presented all the preconditions for the events in the inference to be satisfied.

1. Adam was in the kitchen.
2. Adam wanted something to eat.
3. There was an apple on the kitchen counter.

Figure 3.9: Example narrative with intentional necessity.

In addition to the logical possibility of making the inference, the inference must also be readily available. Hence enablement also requires that the length of the reasoning process to obtain the inferences be relatively small and the alternate choices be few. In the search algorithms used to construct inferences below, the search can is limited to a certain depth or number of states expanded.

The last requirement of enablement is that the elements used in constructing the inference be salient in the reader's mind, as measured by the activation calculation. This will always be the case for inferences that have been recently necessitated. The necessitation of the inference requires that related elements be read in the discourse, and the recency of reading ensures salience. The salience of these elements may decay as other elements are processed, however, and the associated inferences may no longer be enabled.

Necessitation and enablement are defined precisely in the algorithms presented in Sections 3.5.5 and 3.5.6.

3.5.5 Causal Inferencing

Causal inferencing occurs when the reader reasons about how events may occur or may have occurred to produce the current narrative situation. Causal inferences are necessitated when the truth value of propositions changes without an intervening step. To determine if a causal inference is enabled, the current state of the Reader's Story is treated as a planning problem. Given a sufficient set of planning operators to describe the possibilities of events in the narrative, the planner can provide the causal reasoning necessary to determine which steps have occurred in the intervening period. The planner is also able to approximate the difficulty of constructing the inference, by measuring the length of the plan search.

Inferences are most often constructed when the situational possibilities are highly constrained allowing for a relatively small amount of processing to obtain, and a high probability that the inference will prove to be true. In order to make an inference comprehension must necessitate the inferences and the reader must be enabled - via background knowledge or previous narrative clues - to make the inference [72]. Thus, INFER only seeks to construct causal inferences when 1) there is a break in a causal chain in the Reader's Story and 2) the search for a complete sequence of events is short and there are few alternative sets of events. Once the inference is constructed, the salience of the elements related to the

Let $A = L = \emptyset$ and S_r be a sequence of plan steps.

Algorithm CinfNec(A, L, S_r)

1. **Termination** If S_r is empty, return L.
2. **Add Propositions** Let s be the first step of S_r with effects E. Let $A' = A$, $L' = L$, $S_r' = S_r - s$. For each precondition p of s,
   - If $\neg p \in A$ then $L' = L' \bigcup \{p\}$
   - If $\neg p \notin E$ then $A' = A' \bigcup \{p\}$

   For each effect p_e in E $A' = A' \bigcup \{p_e\} - \{\neg p_e\}$
3. **Recursive Invocation** call CinfNec(A', L', S_r')

Figure 3.10: Finding necessitated Causal Inferences.

inference can be checked to satisfy enablement. The search algorithm is given below.

Figure 3.10 displays the algorithm for determining when a causal inference in necessitated. Each step is processed in order, and each of the step's preconditions are checked to against the running list of propositions the reader knows to be true (or false). If the negation of the proposition, p, is in the list, then p has been asserted to be not true in a previous step, and now it is assumed to be true by the execution of this step. Hence, some intervening step that has not been included in the discourse must have occurred, and the proposition p is added to the list of propositions that necessitate inferences. Next, the preconditions and effects of the current step update the running list, and the next step is processed.

The set of preconditions that necessitate inferences is then passed to the algorithm in Figure 3.11. Here, resource bounded partial-order planning is utilized to search for sequences of steps that could have occurred to change the truth value of the proposition p. The planning problem is framed to search only for steps to satisfy p, despite the partial nature of the Reader's Story. However, steps which are added to the plan must have their preconditions satisfied before a solution can be returned. The planning results in a (possibly

Let L be a set of propositions that have changed truth value in the Reader's Story, P_r and $I = \emptyset$

Algorithm CinfEn(L, P_r, I)

1. **Termination** If L is empty, return the elements of I, P_i, that pass the salience check, $SalientInf(P_i, P_r)$.

2. **Planning** Let p be the first proposition of L. $L' = L - \{p\}$. Let P_I be the set of complete plans obtained by conducting planning up to depth d with the initial state as P_r and the open precondition p as the only initial flaw. $I' = I \bigcup P_I$.

3. **Recursive Invocation** call CinfEn(L', P_r, I')

Figure 3.11: Finding enabled Causal Inferences.

$$SalientInf(P_i, S_m) = (\frac{\sum_{s \in P_i} \sum_{e \in s} Act(e, S_m)}{\|s\|}) > c_s$$

Figure 3.12: The salience of an inference is the sum of the activations of its elements.

empty) set of solution plans. These plans are consistent with what the reader knows about the story world and, because the planner was bounded by depth, are easily reachable from the current Reader's Story. The last step is to filter out the inferences which do not have the required salience.

Example

Figure 3.5.5 is the Sequence of Discourse Content for the discourse in Figure 3.8. The sequence is processed one element at a time. First, the proposition Locked(FrontDoor) is added to the Situation Model and the Reader's Story. A new step, step1, with the precondition and effect of Locked(FrontDoor) is created and inserted into the Reader's Story. The plan is checked for necessitated causal inferences with the algorithm in Figure 3.10. Locked(FrontDoor) is added to A, but nothing is added to L. Locked(FrontDoor)

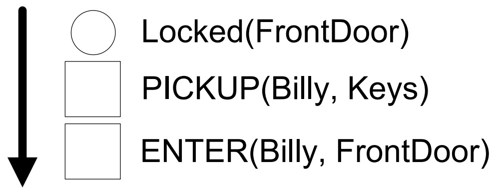

Figure 3.13: Example Sequence of Discourse Content that prompts a causal inference.

does not appear earlier in the plan and, hence, it is not a reversal. Having no necessitated inferences, the MEI model processes the next element.

Next, the step PICKUP(Billy, Keys) is added to the Situation Model and Reader's Story. The new step, step2, is ordered after step1, and the Reader's Story is checked for necessitated inferences with the algorithm in Figure 3.10. Locked(FrontDoor) is added to A, and all the preconditions of PICKUP(Billy, Keys) are checked against A. None of them conflict, so the preconditions and effects of PICKUP(Billy, Keys) are added to A and the algorithm terminates.

Lastly, the step ENTER(Billy, FrontDoor) is added to the Situation Model an Reader's Story. The new step, step3, is ordered after step1 and step2, and the Reader's Story is checked for necessitated inferences with the algorithm in Figure 3.10. Step1 and Step2 are added to A as before, but this time, the precondition of ¬Locked(FrontDoor) is detected to conflict with A. A causal inference is necessitated. The precondition ¬Locked(FrontDoor) of step3 is added to L and returned.

The algorithm in Figure 3.11 is run to determine if an inference is enabled. The planner is invoked with the Reader's Story as the initial state and the Open Precondition ¬Locked(FrontDoor) as the only flaw. The domain for this example contains many possible ways of unlocking a door: actions that represent dynamite, kicking down the door, opening the door from the inside, and unlocking it with a key (UNLOCK-WITH-KEY(?person, ?door, ?key)). The planner searches through all of these possibilities. However, the planner cannot satisfy the preconditions of most of these measures, given what it knows about the story world. These branches result in failure. The planner does find a complete plan within its maximum search depth by using UNLOCK-WITH-KEY(Billy, Keys, FrontDoor) between step2 and step3. This new step is checked for salience, and, since Billy, Keys, and FrontDoor were recently mentioned, the check passes. This inference is both necessitated and enabled. It is predicted by the MEI model.

3.5.6 Intentional Inferencing

Intentional inferencing occurs when the reader predicts character actions that achieve stated character goals. Trabasso and Wiley [14] give evidence that character goals are readily available to readers, whether they are explicitly stated in the text or must be inferred from the characters actions, and that these goals are integral to maintain focus and predict inferences. Readers constantly monitor the goals of the protagonist and related characters to understand why the characters act as they do, improving the cohesion and coherence of the narrative, as in the *coherence assumption* [26]. Readers may also adopt the goals of one or more characters and attempt to address these goals by predicting character actions, as in the *reader goal assumption* [26].

As with causal inferences, intentional inferences are constructed when they are necessitated by the current context. Necessitation is achieved in this case by the statement of an intent. When the reader reads a statement such as "Ellen wanted to buy a new dress." or "Eileen planned to shave her head.", the reader may attempt to infer how the character will achieve the stated goal. The MEI model searches for intentional inferences beginning when an intention is presented in the discourse, and ending when the intention is realized.

Figure 3.14 displays the algorithm for detecting when intentional inferences are necessitated. Each intention that is presented in the discourse is necessitates intentional inferences until it becomes true in the world state.

Figure 3.15 shows the algorithm for determining when intentional inferences are enabled. The set of intentions that have necessitated causal inferences are input. As in causal inference enablement, resource bounded partial-order planning is used to search for sequences of steps to fulfill the desired world state. However, here the actions must be bound to the character which holds the intention. Other than this difference and the initiating intention, the search proceeds the same. The planning problem is framed to search only for steps to satisfy the intention, despite the partial nature of the Reader's Story. Steps which are added to the plan must have their preconditions satisfied before a solution can be returned. The planning results in a (possibly empty) set of solution plans. These plans are consistent with what the reader knows about the story world and, because the planner was bounded by depth, are easily reachable from the current Reader's Story. The last step

Let $A = L = \emptyset$ and S_r be a sequence of plan steps.

Algorithm IinfNec(A, L, S_r)

1. **Termination** If S_r is empty, return L.
2. **Add Propositions** Let s be the first element of S_r, $L' = L$, $A' = A$, $S_r' = S_r - s$.
   - If s is an intention, $L' = L' \bigcup \{s\}$.
   - If s is a step then for each precondition p, if $\neg p \notin E$ then $A' = A' \bigcup \{p\}$, and for each effect p_e, $A' = A' \bigcup \{p_e\} - \{\neg p_e\}$.
3. **Clean Intentions** $L' = L' -$ intentions which represent propositions in A.
4. **Recursive Invocation** call IinfNec(A', L', S_r')

Figure 3.14: Finding necessitated Intentional Inferences, unfulfilled intentions.

Let L be a set of unfulfilled intentions in the Reader's Story, P_r and $I = \emptyset$

Algorithm IinfEn(L, P_r, I)

1. **Termination** If L is empty, return the elements of I, P_i, that pass the salience check, $SalientInf(P_i, P_r)$.
2. **Planning** Let t be the first intention of L, p be the proposition of t and c be the character of t . $L' = L - \{p\}$. Let P_I be the set of complete plans obtained by conducting planning up to depth d with the initial state as P_r, the open precondition p as the only initial flaw, using only actions bound to the character c. $I' = I \bigcup P_I$.
3. **Recursive Invocation** call IinfEn(L', P_r, I')

Figure 3.15: Finding enabled Intentional Inferences.

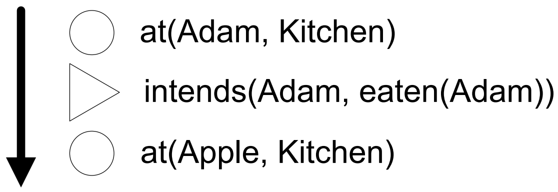

Figure 3.16: Example Sequence of Discourse Content that prompts an intentional inference.

in the algorithm is to filter out the inferences which do not have the required salience.

Example

Figure 3.5.6 is the Sequence of Discourse Content for the discourse in Figure 3.9. As above, the sequence is processed one element at a time. First, the proposition at(Adam, Kitchen) is added to the Situation Model and the Reader's Story. A new step with the precondition and effect of At(Adam, Kitchen) is created and inserted into the Reader's Plan. The plan is checked for necessitated intentional inferences with the algorithm in Figure 3.14. Since there are no intentions, there are no necessitated inferences yet.

The next element, the intention intends(Adam, eaten(Adam)), is processed, added to the Situation Model and the Reader's Story. A frame of commitment is created in the Reader's Story to hold this intention. The algorithm in Figure 3.14 adds the intention to L. Because this is the last element, the intention is not removed from L and an intentional inference is necessitated. The algorithm in Figure 3.15 is invoked to determine if any inferences are enabled. The planner begins with the single Open Intention flaw of eaten(Adam), and plans only with operators for which it can assign Adam to the actor role. The planning domain for this example has an operator EAT(?person, ?food) with the effect of eaten(?person), but the planner is unable to find a complete plan. As of yet, there are no things that are able to be eaten in the Reader's Plan, and the inference is not enabled.

The last element, the proposition at(Apple, Kitchen), is added to the Situation Model and the Reader's Story. A new step is created with the precondition and effect of at(Adam, Kitchen) is created and inserted into the Reader's Story, ordered after the first step and the start of the frame of commitment. The necessity check finds the intention as before, and the planner is invoked for enablement. Now that the Apple has been added to the Reader's Story, the planner finds a complete plan by adding the action EAT(Adam, Apple) after the second step and satisfying all the preconditions with causal links. This new step passes the salience check due to the recent mention of both Adam and the Apple. This inference is both necessitated and enabled. It is predicted by the MEI model.

3.6 Generation

This section presents a partial-order planning algorithm to generate a Discourse Plan by selecting and ordering content from an event log. Generating a Discourse Plan is accomplished by a search in the space of partial plans. Each plan maintains a list of flaws, and children in the search space are generated by choosing a flaw and applying the appropriate resolution method. Causal and Intentional inference flaws require that specific steps be inferred, and these flaws are solved by including elements from the event log to satisfy necessity and enablement criteria.

A Discourse Plan generator creates Discourse Plans from the event log and inferred steps to fulfill the comprehension criteria. The comprehension criteria are satisfied according to the predictions of the reader model, the MEI model in this case, for inferencing necessitated by causal structure, inferencing motivated by intentional structure, and focus. The criteria for causal inferences specify a single step in an inference. To fulfill one of these criterion, the step must be part of a inference that is enabled and causally necessitated, and thus predicted by the reader model. Likewise, the criteria for intentional inferences specify a single step. To fulfill and intentional inference criterion, the step must be part of an inference that is enabled and intentionally necessitated. The criteria for focus specifies a total activation for a node in the Situation Model. To achieve a focus criterion, the specified node must have a total activation at or above the specified amount over the course of the discourse.

3.6.1 Alternate Generation Methods

One method to generate a Discourse Plan with the reader model is by uninformed search. The space of possible Discourse Plans can be searched via a standard search strategy such as breadth first, depth first, or iterative deepening. Children can be generated with a naive method by incrementally adding elements to the Discourse Plan from the event log. At each branch in the tree, a child is created for each remaining element in the event log (repeated states may be combined, since the event log is at least partially ordered).

The reader model is invoked at each node to test for the satisfaction of all comprehension criteria, and the search proceeds until a solution is found or the space is exhausted. The search is sound since it only generates Discourse Plans that include elements from the event log, and it is complete since it searches the entire space of possibilities.

Unfortunately, the space of possible Discourse Plans is exponential in the length of the event log. Let n be the total number of initial conditions, frames of intention, and steps in an IPOCL plan. The size complexity is all the ways to choose a subset of elements, $O(2^n)$. Furthermore, the goal test by the reader model updates the Situation Model, which may be arbitrarily complex and plans with the Reader's Story. Since the planner used for the Reader's Story is resource bound its complexity is constant, though, in practice this computation may take a significant amount of time. Because of these reasons, in my implementation, uninformed search was unable to find solutions to even simple problems with small event logs and short lists of criteria within a reasonable amount of time (one day).

Another approach to generating a Discourse Plan is to use an informed search such as best first or A* with an intelligent heuristic, still generating children with the naive method. Constructing a general purpose heuristic for this problem is non-trivial. The reader model is defined to be context sensitive. Inferences depend not only on the last element read, but possibly all of the previous elements read. The preconditions of the inferred steps must be satisfied from these elements. Focus depends on the relationships between all of the elements in the discourse up until the point of reading. Changing one element in the discourse can have wide ramifications for the comprehension criteria, which may only be revealed later in the search. Thus, an admissible heuristic must be quite conservative in its estimates to the point of being unable to make an educated guess between states. This conservatism makes the speed gains from A* minimal.

Relaxing the admissible requirement, a heuristic for a satisfying best first search faces many of the same problems. For a specified domain, a heuristic can be developed to take advantage of regularities and patterns that emerge between uses. Specific sequences may be useful for prompting inferences or shifting focus, or the author of the domain may have preferences for sequences that may be encoded. However, a general purpose heuristic must differentiate between states where the comprehension effects vary widely with the addition of a single element. Indeed, the last element added in a long sequence of content may satisfy all of the criteria. In the worst case, heuristic search performs no better than uninformed search.

In contrast to heuristics, improving the child function can reliably improve the search results. In the naive case, the child function simply adds every possible element to the Discourse Plan, regardless of how each element may address the comprehension criteria. Some elements may be added that never interact with the criteria, or, worse, become detrimental to satisfying the criteria. However, a more informed child function may directly address each type of criteria. This enhancement prunes the search space from searching combinations of *elements* to searching combinations of *solutions to criteria*. This search is analogous to that of a plan space planner which tries only combinations of steps that address open preconditions, not every combination of steps. If any particular combination of elements does not appear as a solution to one or more comprehension criteria, it will not be in the search space.

3.6.2 INFER Generation

INFER generates Discourse Plans by searching the space of possible Discourse Plans. The root is an empty Discourse Plan, and refinement operators are used to generate children to satisfy flaws in the parent plan. Flaws in the Discourse Plan represent comprehension criteria that have not yet been satisfied or potential consistency problems. The user specifies the initial causal inferencing, intentional inferencing, and focus criteria, and each criteria is converted to a flaw in the discourse plan.

Discourse Plans are generated in partial-order, then linearized to form a Sequence of Discourse Content. A Discourse Plan is an IPOCL plan that contains steps, bindings, causal links, orderings, and frames of commitment. A Discourse Plan may also include inferred steps, which are not included in the final Sequence of Discourse Content. These inferred steps are used as place holders to ensure that inferences are enabled and necessitated. The final Discourse Plan must be constructed such that any linearization is a satisfying Sequence of Discourse Content.

Figure 3.17 is INFER's basic algorithm for generation. INFER selects a flaw and calls the appropriate routine to refine the plan to remove the flaw. The refinement process may create new flaws, such as the open precondition or open effect flaws that are never specified by the user. The search proceeds until a solution is found, a search limit is reached, or the search space is fully exhausted.

Let P be a Discourse Plan, R be an event log, and F be a list of flaws.

Algorithm Refine-Discourse-Plan(P, R, F, Δ)

Parameters: The functions Causal-Inf-Refine (prompt causal inferences), Intentional-Inf-Refine (prompt intentional inferences), Open-Prec-Refine (establish open preconditions), Open-Effect-Refine (link open effects), and Focus-Refine (resolve focus). Each allows for backtracking for other refinements of the same flaw.

1. **Termination Check** If F is empty, return P.

2. **Flaw Selection and Refinement** Let f be either 1) a causal inference flaw, 2) an intentional inference flaw, 3) a link intention flaw, 4) an open precondition flaw, 5) an open effect flaw, or 6) a focus flaw. Switch on the flaw type:

   (a) **Prompt Causal Inference** Let $\langle P', F' \rangle$ = Causal-Inf-Refine(P, R, F, f).

   (b) **Prompt Intentional Inference**
   Let $\langle P', F' \rangle$ = Intentional-Inf-Refine(P, R, F, f).

   (c) **Link to Intention**
   Let $\langle P', F' \rangle$ = Link-to-Intention(P, R, F, f) or Link-to-Intention-NewStep(P, R, F, f, Δ).

   (d) **Establish Open Precondition** Let $\langle P', F' \rangle$ = Open-Prec-Refine(P, R, F, f, Δ) or Open-Prec-Refine-Inf(P, R, F, f, Δ).

   (e) **Link Open Effect** Let $\langle P', F' \rangle$ = Open-Effect-Refine(P, R, F, f, Δ) or Open-Effect-Refine-Inf(P, R, F, f, Δ).

   (f) **Shift Focus** Let $\langle P', F' \rangle$ = Focus-Refine(P, R, F, f).

3. **Consistency Check** If P' does not exist or is inconsistent, backtrack.

4. **Recursive Invocation** Call Refine-Discourse-Plan(P', R, F')

Figure 3.17: Algorithm for refining a Discourse Plan to remove all flaws.

Causal Inference flaws denote steps that are to be causally inferred. Intentional Inference flaws denote steps that are to be intentionally inferred. Link to intention flaws denote steps that are to be causally linked to an intention (for intentional inferencing necessitation). Open precondition flaws denote preconditions of inferred steps that are to be established in the plan (for inferencing enablement). Open effect flaws denote effects of inferred steps that are to be causally linked to non-inferred steps (for causal necessitation). Lastly, focus flaws denote elements which are to have a minimum focus.

Discourse Plans with Causal Inference flaws are refined to include the inferred step as a place holder for further refinement. After the step is inserted, open precondition flaws for each precondition are added and a single open effect flaw is created to ensure the inference is necessitated. If a solution is found further down in the tree, it will have satisfied all of the preconditions of the inferred step and used one of its effects to establish a precondition of a non-inferred step, either directly or through a causal chain.

Discourse Plans with Intentional Inference flaws are also refined to include the inferenced step as a place holder for further refinement. After the step is inserted, open precondition flaws for each precondition are added and a link to intention is created to ensure the inference is necessitated. If a solution is found further down in the tree, it will have satisfied all of the preconditions of the inferred step and used one of its effects to establish a goal of a frame of commitment, either directly or through a causal chain.

Discourse Plans with Link to Intention flaws are refined to create a causal chain between a step and a frame of commitment. The step may be directly linked to the frame of commitment in a single refinement or the refinement may add an inferred step to the causal chain reaching from the frame to the step. If a new step is added, then 1) new Open Precondition flaws are created for each of its preconditions and 2) a new Link to Intention flaw is created to link the original step to the new step. If a solution is found further in the tree, it will have satisfied all of the preconditions of the inferred steps and it will have causally linked the original step to the frame of commitment.

Plans with Open Precondition flaws are refined, similar to the standard manner [8], to add a causal link to the precondition. The link may be created between existing steps in the plan, or a new step may be added at the head of the link. If the new step is taken from the event log, then every ordering with both steps in the Discourse Plan is added to the Discourse Plan to maintain consistency with the event log. If the new step is inferred, then new Open Precondition flaws are created for each of its preconditions. Hence, chains of inferred steps must eventually have their preconditions satisfied by steps from the event log.

Discourse Plans with Open Effect flaws are refined to add a causal link from the effect. The link may be created between existing steps in the plan, or a new step may be added at the tail of the link. As in Open Precondition refinements, If the new step is taken from the event log, then every ordering with both steps in the Discourse Plan is added to the Discourse Plan to maintain consistency with the event log. If the new step is inferred, then 1) new Open Precondition flaws are created for each of its preconditions other than the one being linked and 2) a new Open Effect flaw is created for the new step. Thus, chains of inferred steps must eventually have an effect satisfy a precondition of a non-inferred step.

Discourse Plans with Focus flaws are refined by adding related steps to increase the focus of a particular element. These steps cannot be inferred, and must come from the event log.

3.6.3 Refinement Routines

This section describes each of the routines used by INFER to refine plans and remove flaws. These routines generate the child nodes in the search space. These are the refinement operators for Causal Inference, Open Precondition, Open Effect, Intentional Inference, Link Intention, and Focus flaws, in that order. In the case where there is more than one routine for a single flaw type, INFER may choose among the routines nondeterministically.

Causal Inference Refinements The method for refining plans with Causal Inference flaws is shown in Figure 3.18. A causal inference flaw contains a step, fs, to be inferred and bindings over the free variables in the step. The plan is refined by inserting the step with the bindings as a place holder in the plan for future refinements. After the step has been inserted, INFER must make sure that the inference is necessitated and enabled. One effect, e is chosen from the step to be the proposition that will change without an intervening step, necessitating the inference. An open effect flaw is created for this effect and an open precondition flaw is created for its negation, $\neg e$. These flaws ensure that the inferred step will change the value of the proposition. Flaws are then created for all of the preconditions of the inferred step, ensuring enablement. The reader will read elements with all of the preconditions necessary to infer this step, either directly or through a chain of

Algorithm:

Causal-Inf-Refine($P = \langle S_P, B_P, O_P, L_P, C_P, D_P, I_P \rangle,\ R = \langle S_R, B_R, O_R, L_R, C_R \rangle$,$F$,$f = \langle fs, fb \rangle$)

1. **Create Step** Let s be a new step of type fs with bindings fb.
2. **Create Flaws** Choose an effect e of s and let $f' = \langle s, e \rangle$ be a new open effect flaw. Let f_a be the set of open precondition flaws for each precondition of s union an open precondition flaw for $\neg e$. Let $F' = F - \{f\} \bigcup f_a \bigcup \{f'\}$.
3. **Update Plan** Let $S_P = S_P \bigcup \{s\}$, $D_P = D_P \bigcup \{s\}$, $B_P = B_P \bigcup \{fb\}$.
4. **Consistency Check** Resolve all threats. If s_a does not exist, B_P becomes inconsistent, or the directed graph representing O_P contains a cycle, backtrack.
5. **Return** Return $\langle P, F' \rangle$

Figure 3.18: Algorithm for refining a plan with a causal inference flaw. P is the Discourse Plan, R is the event log, f is the causal inference flaw, fs is the step to be inferred, and fb is the bindings for fs.

inferences.

After a causal inference flaw has been refined, INFER must resolve the open precondition flaws and open effect flaws that arise. These are both resolved by creating a new causal link to another step in the plan from an effect to a precondition. The other step can be obtained by 1) reusing a step in the Discourse Plan, 2) taking an unused step from the event log, or 3) creating a new inferred step. In the case of reusing a step, the causal link is merely created, and the plan is checked for consistency. When taking an unused step from the event log, INFER maintains the ordering and causality of steps in the event log by transporting not only the step, but all of the causal links and orderings that have both their head and tail in the Discourse Plan. When INFER creates a new inferred step, new flaws are created to ensure the step's preconditions are fulfilled and to ensure at least one of the step's effects become linked.

Figure 3.19 refines an open precondition flaw by either reusing a step in the Dis-

Algorithm: P is a Discourse Plan, R is an event log, F is a set of plan flaws, f is an open precondition flaw, fs is the step with the open precondition, and p is the precondition.
Open-Precondition-Refine($P = \langle S_P, B_P, O_P, L_P, C_P, D_P, I_P \rangle$, $R = \langle S_R, B_R, O_R, L_R, C_R \rangle$,$F$,$f = \langle fs, p \rangle$,$\Delta$)

1. **Choose Step** Let s_a be a step from $S_P \bigcup S_R$ with a non-cancelled precondition or effect that unifies with p with bindings b_a. Let l_a be a causal link from s_a to s over p.

2. **Update Plan** Let $S_P = S_P \bigcup \{s_a\}$, $B_P = B_P \bigcup \{b_a\}$, $O_P = O_P \bigcup \{s_a \rightarrow s\}$, $L_P = LP \bigcup \{l_a\}$.

3. **Consistency Check** Resolve all threats. If B_P becomes inconsistent, or the directed graph representing O_P contains a cycle, backtrack.

4. **Return** Return $\langle P, F - \{f\} \rangle$

Figure 3.19: Algorithm for refining a plan with an open precondition flaw. This is a standard partial-order planning technique [8].

course Plan or taking an unused step from the event log. Open precondition flaws occur when a precondition of an inferred step is unsatisfied. If a step is reused, a causal link is formed and an ordering is created. The plan is checked for consistency and returned, the open precondition flaw removed. If a step is taken from the event log, then the step is added, a casual link is formed, and an ordering is created. Causal links and orderings concerning the new step are transferred to the Discourse Plan, when their heads and tails are both in the Discourse Plan. The plan is checked for consistency and returned; the open precondition flaw is removed.

Figure 3.20 refines an open precondition flaw by creating a newly inferred step from the action library. The new step is created, the bindings are added that unify with the open precondition, a causal link is created, and an ordering is added. Since the new step is inferred it will need to have its preconditions satisfied, and new open precondition flaws are created for each. The instigating open precondition flaw is removed, and the plan is checked for consistency and returned.

Figure 3.21 is the routine for refining an open effect flaw by either reusing a step

Algorithm: P is a Discourse Plan, R is an event log, F is a set of plan flaws, f is an open precondition flaw, fs is the step with the open precondition, and p is the precondition. This routine is only applied to open preconditions of inferred steps.

Open-Precondition-Refine-Inf($P = \langle S_P, B_P, O_P, L_P, C_P, D_P, I_P \rangle$, $R = \langle S_R, B_R, O_R, L_R, C_R \rangle$,$F$,$f = \langle fs, e \rangle$,$\Delta$)

1. **Choose Post Step** Let s_a be a new step from Δ with an effect that unifies with p with bindings b_a. Let l_a be a causal link from s_a to s over p.

2. **Create Flaws** Let f_a be the set of open precondition flaws for each precondition of s_a. Let $F' = F - \{f\} \bigcup f_a$.

3. **Update Plan** Let $S_P = S_P \bigcup \{s_a\}$, $B_P = B_P \bigcup \{b_a\}$, $O_P = O_P \bigcup \{s_a \rightarrow s\}$, $L_P = LP \bigcup \{l_a\}$.

4. **Consistency Check** Resolve all threats. If s_a does not exist, B_P becomes inconsistent, or the directed graph representing O_P contains a cycle, backtrack.

5. **Return** Return $\langle P, F' \rangle$

Figure 3.20: Algorithm for refining a plan with an open precondition flaw where the precondition is part of an inferred step, and the satisfying step is also inferred.

Algorithm: P is a Discourse Plan, R is an event log, F is a set of plan flaws, f is an open effect flaw, fs is the step with the open effect, and e is the effect.

Open-Effect-Refine($P = \langle S_P, B_P, O_P, L_P, C_P, D_P, I_P \rangle$, $R = \langle S_R, B_R, O_R, L_R, C_R \rangle$,$F$,$f = \langle fs, e \rangle$,$\Delta$)

1. **Choose Post Step** Let s_a be a step from $S_P \bigcup S_R$ with a precondition that unifies with e with bindings b_a. Let l_a be a causal link from s to s_a over e.

2. **Update Plan** Let $S_P = S_P \bigcup \{s_a\}$, $B_P = B_P \bigcup \{b_a\}$, $O_P = O_P \bigcup \{s \rightarrow s_a\}$, $L_P = LP \bigcup \{l_a\}$.

3. **Consistency Check** Resolve all threats. If s_a does not exist, B_P becomes inconsistent, or the directed graph representing O_P contains a cycle, backtrack.

4. **Return** Return $\langle P, F - \{f\} \rangle$

Figure 3.21: Algorithm for refining a plan with an open effect flaw where the effect is taken from the event log.

in the Discourse Plan or taking an unused step from the event log. Open effect flaws occur when a step is causally inferred, and requires one of its effects to be linked to necessitate the inference. If a step is reused, a causal link is formed and an ordering is created. The plan is checked for consistency and returned, the open precondition flaw removed. If a step is taken from the event log, then the step is added, a casual link is formed, and an ordering is created. Causal links and orderings concerning the new step are transferred to the Discourse Plan, when their heads and tails are both in the Discourse Plan. The plan is checked for consistency and returned; the open effect flaw is removed.

Figure 3.20 refines an open effect flaw by creating a newly inferred step from the action library. The new step is created, the bindings are added that unify with the open effect, a causal link is created, and an ordering is added. Since the new step is inferred it will need to have its preconditions satisfied, and new open precondition flaws are created for each. The effect for which the flaw was created has not yet been linked via causal chain to a non-inferred step, and it will not yet be visible to the reader. A new open effect flaw is created for the added step to continue the causal chain until a non-inferred step is reached.

Algorithm: P is a Discourse Plan, R is an event log, F is a set of plan flaws, f is an open effect flaw, fs is the step with the open effect, and e is the effect.

Open-Effect-Refine-Inf($P = \langle S_P, B_P, O_P, L_P, C_P, D_P, I_P \rangle$, $R = \langle S_R, B_R, O_R, L_R, C_R \rangle$,$F$,$f = \langle fs, e \rangle$,$\Delta$)

1. **Choose Post Step** Let s_a be a new step from Δ with a precondition that unifies with e with bindings b_a. Let l_a be a causal link from s to s_a over e.

2. **Create Flaws** Let f_a be the set of open precondition flaws for each precondition of s_a other than e. Choose an effect e' of s_a and let $f' = \langle s_a, e' \rangle$ be a new open effect flaw. Let $F' = F - \{f\} \bigcup f_a \bigcup \{f'\}$.

3. **Update Plan** Let $S_P = S_P \bigcup \{s_a\}$, $B_P = B_P \bigcup \{b_a\}$, $O_P = O_P \bigcup \{s \rightarrow s_a\}$, $L_P = LP \bigcup \{l_a\}$.

4. **Consistency Check** Resolve all threats. If s_a does not exist, B_P becomes inconsistent, or the directed graph representing O_P contains a cycle, backtrack.

5. **Return** Return $\langle P, F' \rangle$

Figure 3.22: Algorithm for refining a plan with an open effect flaw where the effect is part of an inferred step.

The instigating open effect flaw is removed, and the plan is checked for consistency and returned.

Intentional Inference Refinements Intentional inferences are refined in a method similar to causal inferences, but the steps to build a causal chain between inferred step and frame of commitment are different. First, a place holder inferred step is created to contain the inference, and a frame of commitment is chosen to necessitate the inference. A new link to intention flaw is created to create a causal chain between the two. The link to intention flaw is resolved either by creating a causal link from the step to the frame of commitment, or by linking the step to a new inferred step (extending the causal chain) and creating a new link to intention flaw.

Figure 3.23 is the routine for refining an intentional inference flaw. A new inferred

Algorithm:

Intentional-Inf-Refine$(P = \langle S_P, B_P, O_P, L_P, C_P, D_P, I_P \rangle,$
$R = \langle S_R, B_R, O_R, L_R, C_R \rangle,\ F,\ f = \langle fs, fb \rangle)$

1. **Create Step, Choose a Frame of Commitment** Let s be a new step of type fs with bindings fb. Choose a frame of commitment c with character d as the actor from $C_P \bigcup C_R$.

2. **Create Flaws** Let f_a be the set of open precondition flaws for each precondition of s Choose an effect e of s and let $f' = \langle s, e, c \rangle$ be a new link intention flaw. Let $F' = F - \{f\} \bigcup f_a \bigcup \{f'\}$.

3. **Update Plan** $S_P = S_P \bigcup \{s\}$, $I_P = I_P \bigcup \{s\}$, $C_P = C_P \bigcup \{c\}$, $B_P = B_P \bigcup \{fb\}$, include s in c.

4. **Consistency Check** Resolve all threats. If B_P becomes inconsistent, or the directed graph representing O_P contains a cycle, backtrack.

5. **Return** Return $\langle P, F' \rangle$

Figure 3.23: Refining a plan with an intentional inference flaw.

step is instantiated with the character d as the actor, and a frame of commitment with the same character is chosen. New open precondition flaws are created for the new step. An effect is chosen and a new link intention flaw is created to link the effect along a causal chain to the goal of the frame of commitment. The plan is checked for consistency, the intentional inference flaw is removed, and the new plan is returned.

Figure 3.24 is the routine for refining a link to intention flaw where the step can be directly linked to the frame of commitment or step in the causal chain. In this case, a new causal link is created, new bindings may be added, and an ordering is added. The plan is checked for consistency, the instigating link to intention flaw is removed, and the plan is returned.

Figure 3.25 is the routine for refining a link to intention flaw where the causal chain to the intention is extended, backwards, with another inferred step. A new inferred

Algorithm:

Link-to-Intention$(P = \langle S_P, B_P, O_P, L_P, C_P, D_P, I_P \rangle,$
$R = \langle S_R, B_R, O_R, L_R, C_R \rangle$,$f = \langle fs, fc, d \rangle$, $\Delta)$

1. **Choose Goal**
   - **If fc is a frame of commitment:** Let g be the goal of fc.
   - **If fc is a step:** Choose g to be a precondition of fc.
2. **Link to Directly to Goal** Choose an effect e of fs that unifies with g with bindings b.
3. **Update Plan** Let $B_P = B_P \bigcup \{b\}$, , $O_P = O_p \bigcup \{(fs \rightarrow fc)\}$, $L_P = L_P \bigcup \{fs \xrightarrow{e} fc\}$.
4. **Consistency Check** If B_P becomes inconsistent or the directed graph representing O_P contains a cycle, backtrack.
5. **Return** Return $\langle P, F - \{f\} \rangle$

Figure 3.24: Linking an intentional inference to a frame of commitment goal. Δ is the action library of possible steps, fs is the step to be inferred, and fc is the frame of commitment with character d

step is created to either link directly to the goal of the frame of commitment or to the first step of the causal chain. A causal link and an ordering are formed, and any new bindings are added. New open precondition flaws are created for the new inferred step, and the instigating flaw is removed. To continue the causal chain back to the original step, a new link to intention is formed to link the original step to the newly inferred step. The plan is checked for consistency, and returned.

Focus Refinements After the inference refinements are completed, INFER addresses the focus flaws to ensure that elements in the Discourse Plan receive the desired amount of focus. Figure 3.26 is the algorithm for refining plans with focus flaws. An unused step from the event log that includes the element to be focused is chosen. The step is inserted into the Discourse Plan. As in the inference refinements, each relevant causal link and ordering from the event log is transported to the Discourse Plan. The activation of the element over the entire plan is computed. If the activation is above the required value, then the flaw is removed. Else, the flaw is kept for another round of refinement. Finally, the plan is checked for consistency, and returned.

3.6.4 Threat Detection and Resolution

The Discourse Plan is partially ordered, and the effects of causal links may be threatened to be undone by steps that may intervene. This occurance is referred to as a threat [8]. Discourse Plans have two types of causal links, exclusive and non-exclusive. A exclusive link is one where the head step must be the sole cause of the effect. Links are exclusive for inferencing purposes to keep other steps from destroying the necessitation of the inference. Links created by open effect flaw refinements and link to intention flaw refinements are exclusive. All other links are non-exclusive. An non-exclusive link is one where the head step may be one of many causes of the effect (however, only one causal link is allowed for each precondition). Links created by open precondition flaw refinements and links transported from the event log are non-exclusive.

Figure 3.27 is the rule for detecting a threat. Threats occur for non-exclusive causal links when an intervening step's effect negates the link's proposition. Threats occur for exclusive links when an intervening step's effect equals or negates the link's proposition. These threats are resolved before the plan is added to the search space. If the threat cannot be resolved, then the search must backtrack.

Algorithm:

Link-to-Intention-NewStep$(P = \langle S_P, B_P, O_P, L_P, C_P, D_P, I_P \rangle,$
$R = \langle S_R, B_R, O_R, L_R, C_R \rangle, f = \langle fs, fc, d \rangle,\ \Delta)$

1. **Choose Effect and Goal** Choose an effect e of fs.
   - **If fc is a frame of commitment:** Let g be the goal of fc.
   - **If fc is a step:** Choose g to be a precondition of fc.
2. **Add New Step and Link** Let s_{add} be a newly instantiated step from Δ with character d and effect e_{add} that unifies with g with bindings b_{add}.
3. **Create Flaw** Let $f' = \langle fs, s_{add}, d \rangle$ be a new link intention flaw and $F' = F - \{f\} \bigcup \{f'\}$.
4. **Update Plan** Let $S_p = S_p \bigcup \{s_{add}\}$, $B_P = B_P \bigcup \{b_{add}\}$, $O_P = O_p \bigcup \{s_{add} \rightarrow fc)\}$, $D_P = D_P \bigcup \{s_{add}\}$, $L_P = L_P \bigcup \{s_{add} \xrightarrow{e} fc\}$.
5. **Consistency Check** If s_{add} does not exist, B_P becomes inconsistent, or the directed graph representing O_P contains a cycle, backtrack.
6. **Return** Return $\langle P, F' \rangle$

Figure 3.25: Linking an intentional inference to a frame of commitment goal. Δ is the action library of possible steps, fs is the step to be inferred, and fc is the frame of commitment with character d

Algorithm: P is a Discourse Plan, R is an event log, F is a set of plan flaws, f is an open precondition flaw, fs is the step with the open precondition, and p is the precondition.
Focus-Refine($P = \langle S_P, B_P, O_P, L_P, C_P, D_P, I_P\rangle$, $R = \langle S_R, B_R, O_R, L_R, C_R\rangle$,$F$,$f = \langle e, x\rangle$,$\Delta$)

1. **Check Focus** Construct the Situation Model from the entire Discourse Plan, and total the activation of e. If the total is above x then let $F' = F - \{f\}$ and Return. Else, let $F' = F$.

2. **Choose Step** Let s_a be a step from $S_R - S_P$ with the bindings b_a and a Situation Model graph containing the node e.

3. **Update Plan** Let $S_P = S_P \bigcup \{s_a\}$, $B_P = B_P \bigcup \{b_a\}$.

4. **Consistency Check** Resolve all threats. If s_a does not exist, B_P becomes inconsistent, or the directed graph representing O_P contains a cycle, backtrack.

5. **Return** Return $\langle P, F'\rangle$

Figure 3.26: Algorithm for refining a plan with a focus flaw.

Let l be a causal link $s_i \xrightarrow{p} s_j$. The link is threatened if

- There is a step s_k be a step with an effect $\neg e$ and possibly $s_i \rightarrow s_k \rightarrow s_j$.

- l is a exclusive link and there is a step s_k be a step with an effect e and possibly $s_i \rightarrow s_k \rightarrow s_j$.

Figure 3.27: Rule for detecting a threat.

Let s_k be a step threatening the link $s_i \xrightarrow{p} s_j$ with the effect e.

1. **Resolve Threat** Choose one of the following:
   (a) **Promotion** Let $O_P = O_P \bigcup \{s_j < s_k\}$
   (b) **Demotion** Let $O_P = O_P \bigcup \{s_k < s_i\}$
   (c) **Separation** Let b be a set of bindings such that e no longer unifies with p, let $B_P = B_p \bigcup \{b\}$
2. **Consistency Check** If B_P becomes inconsistent or the directed graph representing O_P contains a cycle, backtrack.
3. **Return** P

Figure 3.28: Resolving a threaten link.

Figure 3.28 is the algorithm for resolving a threat to a causal link [8]. The intervening step can be promoted by ordering after the tail step. It can be demoted by ordering it before the head step. Or, it can be separated by adding binding constraints such that the bindings no longer have a possible unification.

3.6.5 Pruning and Heuristics

The prompting of inferences by the Causal Inference and Intentional Inference flaw refinements ensures the necessitation and logical enablement of the inference. The flaws do not directly address the length of the reasoning process used to construct the inference, another facet of enablement. The length of the causal chain that embodies the inference must be monitored as it is constructed. An inference chain which is too long, or too difficult to construct is not an inference that is likely to be constructed by the reader. Since the causal chains only grow as the search proceeds, once the inference can be deemed too difficult to construct, that Discourse Plan can be pruned from the search space. The inferencing of the MEI model may be invoked to determine when this threashold has been reached.

Constructing a heuristic for the search of Discourse Plans is a non-trivial task.

Depending on the size and structure of the event log, the branching factor may be very large, and the heuristic must choose between a set partial Discourse Plans with few distinctions between them. As is the intuition in A* search, the heuristic may combine the current 'cost' of the plan with the estimated 'cost' of completing the plan. The current cost may be estimated in length, shorter discourses that fulfill the same criteria may be preferred. The estimated cost may be a function of the number of remaining flaws weighted by their type. A sum of these two weights balances the current length with the expectancy of completing a particular branch in the plan space, and does tend to lead to finding a solution sooner. For simplicity and regularity, a simple breadth first search was used in generating the Discourse Plans in the experiments.

The order in which the flaws are chosen for refinement and the choice of steps for satisfying flaws may have a large impact on the length of the search process. Generally, it is prudent to choose the flaw that will impose the most constraints on the search space at any one time. This choice will narrow the possibilities for the rest of the search, shrinking the search space going forward. The inference criteria seem to be the most constraining when refined, due to the requirements of the open preconditions and open effects or link to intention flaws. When choosing steps to satisfy open preconditions or effects, it is often beneficial to choose steps with the most preconditions and effects overlapping with the current set of flaws. Depending on the structure of the event log and planning domain, these steps have the potential to satisfy more flaws by their inclusion.

3.7 Example

This section presents a detailed example to illustrate the operation of INFER. First, the domain author must define the inputs. An event log of the happenings of a wild west town leading up to a bank robbery serve as the story for this example; Figure 3.29 list some of the events of this story. An operator library for this domain is included, defining templates for events in this wild west town. Next, the domain author must choose the comprehension criteria that INFER's planner will attempt to satisfy. Examining the story, the domain author decides upon the following comprehension goals.

1. The reader should causally infer that Robbie pickpocketed Sally.

2. The reader should intentionally infer that Robbie will hold up the bank.

- ...
- Intends (Sally (has Sally Bluedress))
- WITHDRAW-MONEY (Sally Bank)
- PICKPOCKET (Robbie Sally Money1 Mainstreet Darkalley)
- REPORT-STOLEN (Sally Sheriff Money)
- BUY-DRINKS-FOR (Robbie Barney Money1)
- LAY-TO-DRUNKEN-SLEEP (Robbie Barney Darkalley)
- TAKE-THING-OFF-SLEEPER (Robbie Barney Sixshooter Darkalley)
- ...

Figure 3.29: A selection of some of the events in the story. Events leading up to a bank robbery in a western town. Each event represents an IPOCL plan step or frame of commitment.

3. The reader should focus on the Six Shooter gun for at least 2-3 steps.

These criteria lead the domain author to construct three corresponding Discourse Plan flaws: 1) a causal inference flaw for the step (PICKPOCKET Robbie Sally Money1 Mainstreet Darkalley), 2) an intentional inference flaw for the step (HOLD-UP-BANK Robbie Sixshooter Bank), and 3) a focus flaw for Sixshooter at the total of 1.0 activation. These flaws, the story, and the operator library are input into INFER, and the search for a Discourse Plan proceeds as follows.

In resolving the causal inference flaw, INFER follows the routine in figure 3.18. The pickpocket operator is instantiated into a new step, PICKPOCKET-1, with the bindings (robbie sally money1 mainstreet darkalley) for the variables (?person ?mark ?money ?place ?alley), respectively. Figure 3.30 is the abstract plan operator for pickpocket steps. In this domain, a thief may pickpocket a victim (a.k.a. mark) if the victim is at a particular location, the thief is hidden in an alley nearby, and the victim has some money. Figure 3.31 is the instantiation of this operator using the specified bindings. The action results in the mark no longer having having the money, and the thief having the money. The result is shown in figure 3.31 Open precondition flaws are created for each of the preconditions

```
(define (action pickpocket)
    :parameters (?person ?mark ?money ?place ?alley)
    :actors (?person)

    :precondition ((person ?person) (evil ?person) (place ?place)
                   (place ?alley) (alley ?alley)
                   (alley-of ?alley ?place)
                   (person ?mark) (money ?money))

                   (at ?person ?alley) (at ?mark ?place)
                   (hidden ?person) (has ?mark ?money))

    :effect ((has ?person ?money) (:not (has ?mark ?money))))
```

Figure 3.30: The 'pickpocket' operator.

```
(define (action pickpocket-1)
    :parameters (robbie sally money1 mainstreet darkalley)
    :actors (robbie)

    :precondition ((person robbie) (evil robbie) (place mainstreet)
                   (place darkalley) (alley darkalley)
                   (alley-of darkalley mainstreet)
                   (person sally) (money money1))

                   (at robbie darkalley) (at sally mainstreet)
                   (hidden robbie) (has sally money1))

    :effect ((has robbie money1) (:not (has sally money1))))
```

Figure 3.31: The 'pickpocket' operator instantiated with the bindings (robbie sally money1 mainstreet darkalley).

PICKPOCKET-1 (Robbie Sally Money1 Mainstreet Darkalley)**

Figure 3.32: The Discourse Plan after resolving the causal inference flaw for PICKPOCKET-1. Current flaws: *Open Precondition*: (person robbie) (evil robbie) (place mainstreet) (place darkalley) (alley darkalley) (alley-of darkalley mainstreet) (person sally) (money money1) (at robbie darkalley) (at sally mainstreet) (hidden robbie) (has sally money1). *Open Effect:* PICKPOCKET-1. *Intentional Inference:* HOLD-UP-BANK. *Focus:* SIXSHOOTER.

of the new step; the reader must know that each of these preconditions are true before the inference can be made. In this case, the reader must know facts such as Robbie is a person, Robbie is evil, Main Street is a place, Robbie is at the Dark Alley, Sally is at Main Street, and Sally has the Money. Next, one of the effects is selected to serve as the open effect flaw. Here, the planner chooses the effect (:not (has sally money1)). Thus, some event will be revealed to the reader that indicates that Sally no longer has the Money. The plan is checked for consistency, and the check passes. The causal inference flaw is removed, and the planner chooses the next flaw to address. The partial plan after this refinement is shown in Figure 3.32.

The planner chooses PICKPOCKET-1's precondition (person robbie) as the next open precondition to address. Using the routine in figure 3.19, the planner searches for a step with this precondition or effect. A step from the Discourse Plan might be reused, but since the Discourse Plan is empty the planner must include a step from the story. A HATCH-PLAN step in the story has the precondition (person robbie) and it is selected for inclusion. The step's parameters are bound as (HATCH-PLAN robbie sixshooter brownhorse bank big-money) - indicating that Robbie hatches a plan to use the Six Shooter and the Brown Horse to rob the Bank and take the large amount of Money. This step is instantiated in the Discourse Plan as HATCH-PLAN-2. One of the preconditions to this step is (person robbie), and a new link is created from this step and precondition to PICKPOCKET-1. Next, all of the causal links and orderings with both ends in the Discourse Plan are transferred from the story. Since HATCH-PLAN was one of the first steps in the story, causal links are transferred that link from the initial state to HATCH-PLAN-2, establishing some of the preconditions. However, because HATCH-PLAN-2 is not an inferred step, no new open precondition flaws are created. The plan is checked for consistency, and the check passes. The open precondition flaw is removed. The partial plan after this refinement is shown in

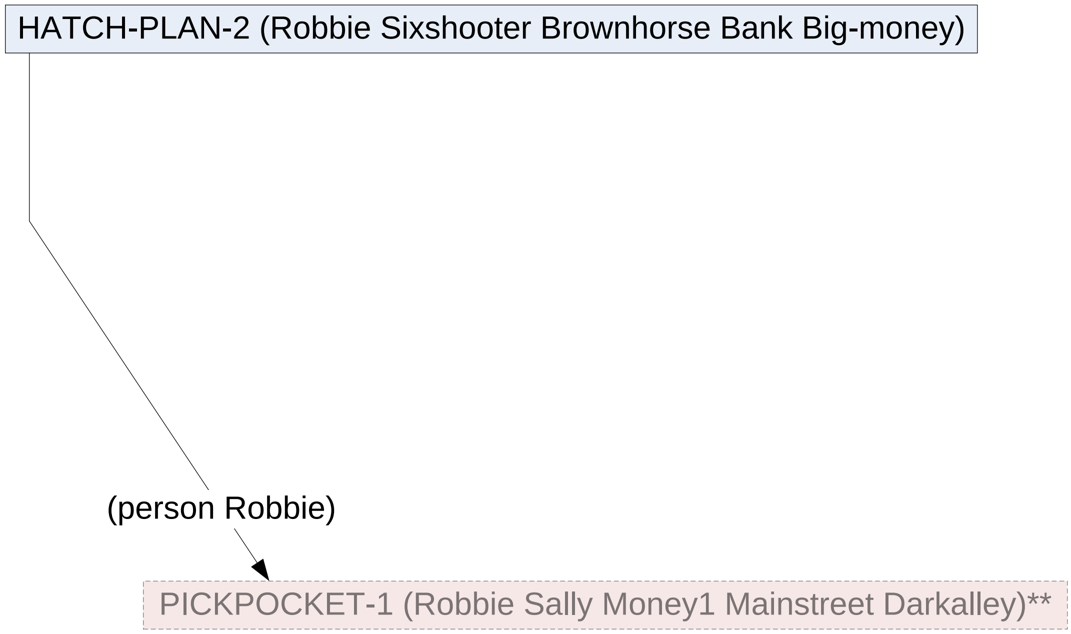

Figure 3.33: The Discourse Plan after resolving the open precondition flaw for (person robbie) for PICKPOCKET-1 (initial state and links not shown). Earlier additions are lightly shaded in this figure. Current flaws: *Open Precondition*: (evil robbie) (place mainstreet) (place darkalley) (alley darkalley) (alley-of darkalley mainstreet) (person sally) (money money1) (at robbie darkalley) (at sally mainstreet) (hidden robbie) (has sally money1). *Open Effect:* PICKPOCKET-1. *Intentional Inference:* HOLD-UP-BANK. *Focus:* SIXSHOOTER.

Figure 3.33.

The planner chooses (evil robbie) as the next open precondition flaw to address. Since HATCH-PLAN-2 contains the precondition (evil robbie), the planner selects this step for reuse. A new causal link is created from HATCH-PLAN-2 to PICKPOCKET-1 over the precondition (evil robbie), and the corresponding open precondition flaw is removed. The partial plan resulting from this refinement is shown in Figure 3.34.

Continuing with open preconditions, the planner chooses (place mainstreet) as the next flaw. In a similar manner as above, the story step (MOVE-TO-ALLEY robbie mainstreet darkalley), indicating that Robbie moves from Mainstreet and hides in the Dark Alley, is chosen to fulfill this precondition. The step is instantiated in the discourse as MOVE-TO-ALLEY-3 and ordered after HATCH-PLAN-2 to correspond with an ordering in the story. The current flaw is removed and the planner continues choosing open precondition flaws to address. MOVE-TO-ALLEY-3 proves to be quite effective for this purpose and is

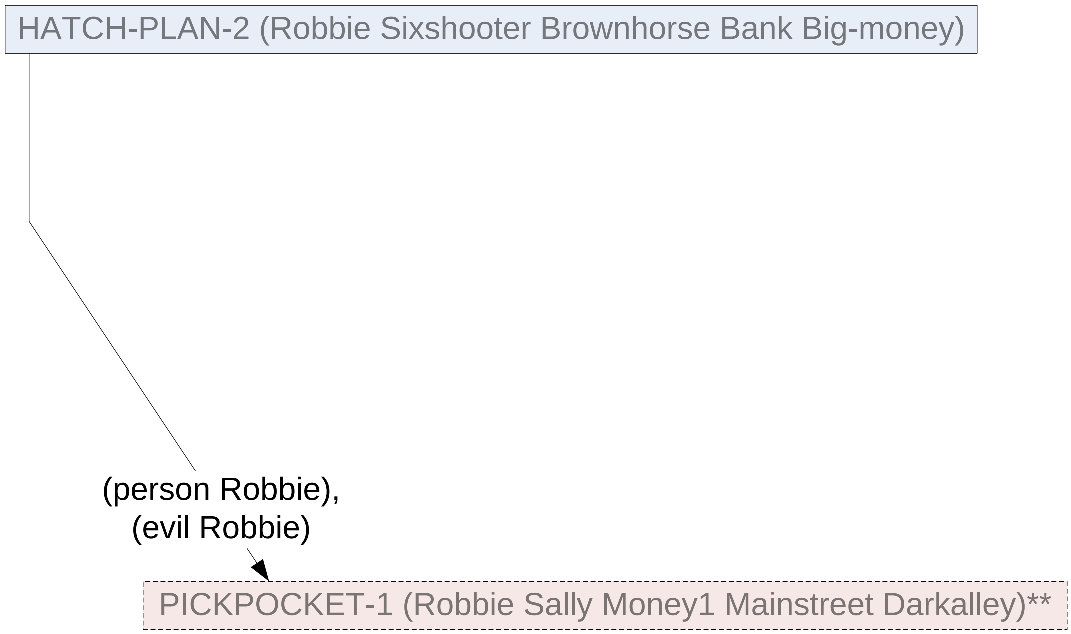

Figure 3.34: The Discourse Plan after adding resolving another open precondition flaw for (evil robbie) for PICKPOCKET-1. Earlier additions are lightly shaded in this figure. Current flaws: *Open Precondition*: (place mainstreet) (place darkalley) (alley darkalley) (alley-of darkalley mainstreet) (person sally) (money money1) (at robbie darkalley) (at sally mainstreet) (hidden robbie) (has sally money1). *Open Effect:* PICKPOCKET-1. *Intentional Inference:* HOLD-UP-BANK. *Focus:* SIXSHOOTER.

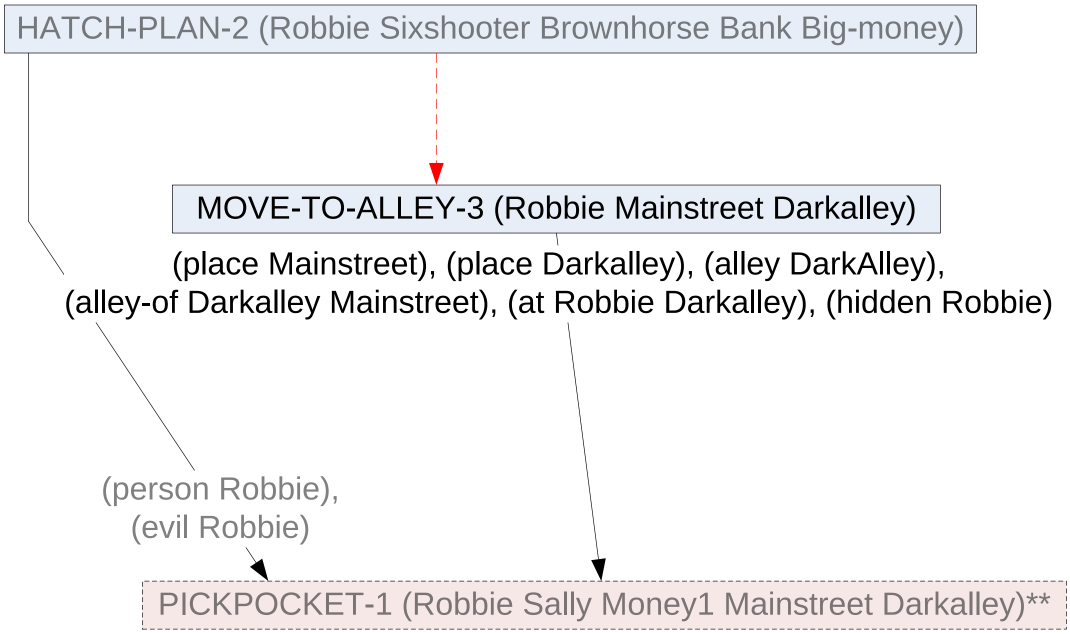

Figure 3.35: The Discourse Plan after resolving some open precondition flaws for PICKPOCKET-1. Earlier additions are lightly shaded in this figure. Current flaws: *Open Precondition*: (person sally) (money money1) (at sally mainstreet) (has sally money1). *Open Effect:* PICKPOCKET-1. *Intentional Inference:* HOLD-UP-BANK. *Focus:* SIXSHOOTER.

reused to establish (place mainstreet), (place darkalley), (alley darkalley), (alley-of darkalley mainstreet), (at robbie darkalley), and (hidden robbie). Causal links are created for each of these as the Discourse Plan is refined as shown in Figure 3.35.

The remaining open precondition flaws concern Sally and the Money: (person sally), (money money1), (has sally money1), (at sally mainstreet). The story step (WITHDRAW-MONEY SALLY BANK MONEY1) indicating that Sally withdrew some Money from the Bank is selected to establish (person sally) is selected and instantiated as WITHDRAW-MONEY-4. No causal links or orderings are transferred from the story concerning this step, either the heads or tails are currently missing from the Discourse Plan. Then, WITHDRAW-MONEY-4 is also used to establish (money money1) and (has sally money1). The last open precondition, (at sally mainstreet), is established by instantiating the story step (WALK-TO SALLY BANK MAINSTREET) as WALK-TO-5. WALK-TO-5 is ordered after WITHDRAW-MONEY-4, as per the ordering in the story. The plan after refining all of the causal inference's open precondition flaws is displayed in Figure 3.36.

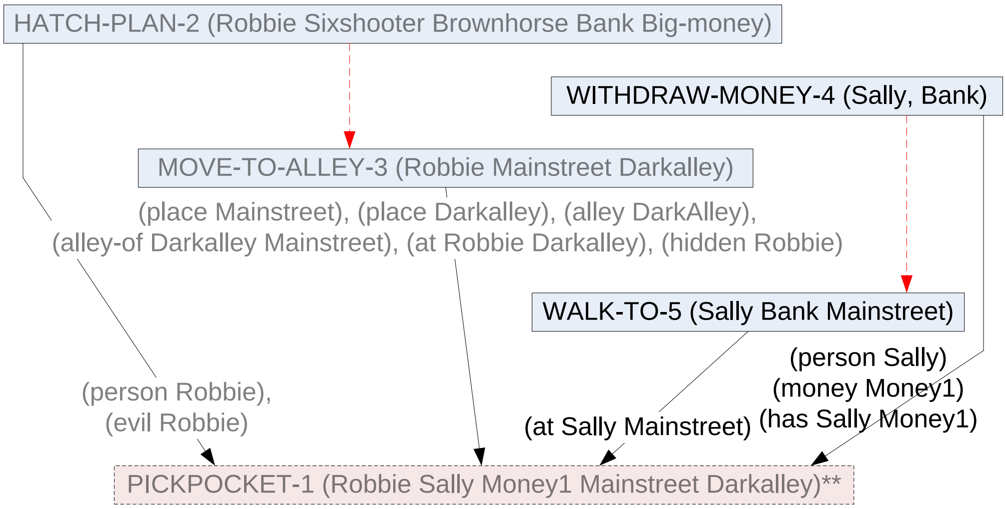

Figure 3.36: The Discourse Plan after refining all of the open precondition flaws of PICKPOCKET-1. Earlier additions are lightly shaded in this figure. Current flaws: *Open Effect:* PICKPOCKET-1. *Intentional Inference:* HOLD-UP-BANK. *Focus:* SIXSHOOTER.

After the open precondition flaws have been addressed, the open effect flaw for (:not (has sally money1)) is selected for refinement. Either of the routines in Figure 3.21 or Figure 3.22 are applicable, but the current heuristic dictates that reusing steps from the story and discourse is preferable to creating new inferred steps. Because there is currently no step in the Discourse Plan with the desired precondition, the story step (FAIL-BUY-DRESS SALLY DRESS DRESSSHOP MONEY1) - indicating that Sally attempts to buy a Dress at the Dress Shop with the Money, but fails due to lack of the Money - is selected for instantiation in the Discourse Plan as FAIL-BUY-DRESS-6. FAIL-BUY-DRESS-6 is ordered after WALK-TO-6 and WITHDRAW-MONEY4 as per orderings in the story. The resulting plan is displayed in Figure 3.37. The refinement of all the flaws resulting from the causal inference flaw is now complete, and the planner proceeds with the intentional inference flaw.

The next flaw is an intentional inference flaw for the step (HOLD-UP-BANK robbie sixshooter bank) - denoting that Robbie holds up the Bank with the Six Shooter is to be inferred via intentional necessitation. The routine in Figure 3.23 is used to refine the plan. First, the new inferred step is created and inserted. Figure 3.38 is the abstract plan operator for the HOLD-UP-BANK action, and Figure 3.39 is the instantiation of this step, HOLD-UP-BANK-7. Next, a frame of commitment is chosen for inclusion in the Discourse

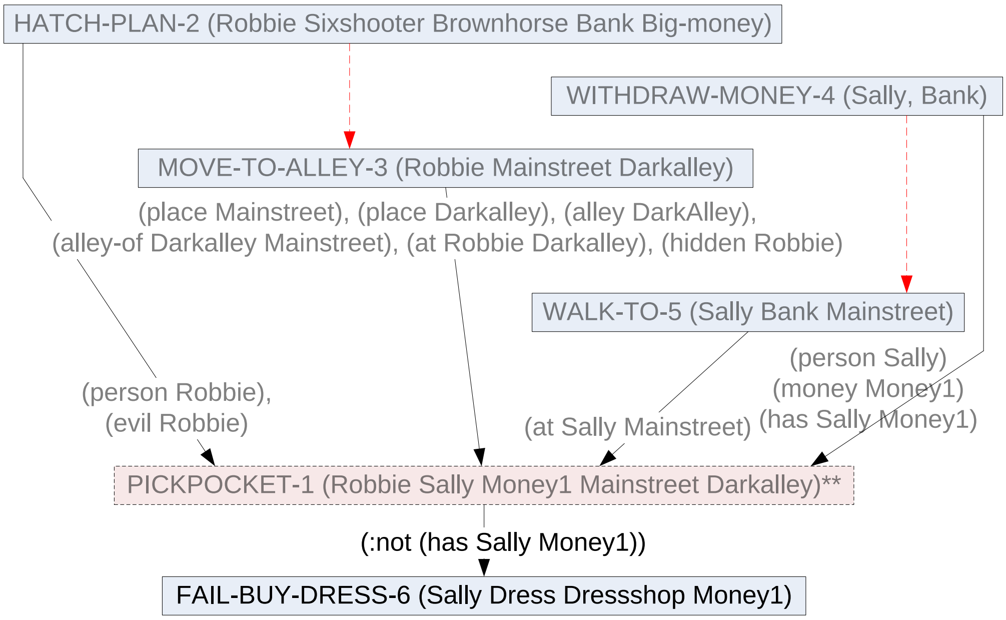

Figure 3.37: The Discourse Plan after resolving the open effect flaw for PICKPOCKET-1. Earlier additions are lightly shaded in this figure. Current flaws: *Intentional Inference:* HOLD-UP-BANK. *Focus:* SIXSHOOTER.

```
(define (action hold-up-bank)
    :parameters (?person ?gun ?bank)
    :actors (?person)
    :precondition ((person ?person) (evil ?person) (bank ?bank)
                   (gun ?gun)
                   (at ?person ?bank) (has ?person ?gun))
    :effect ((held-up ?person ?bank)))
```

Figure 3.38: The 'hold-up-bank' operator.

```
(define (action hold-up-bank-7)
    :parameters (robbie sixshooter bank)
    :actors (robbie)
    :precondition ((person robbie) (evil robbie) (bank bank)
                   (gun sixshooter)
                   (at robbie bank) (has robbie sixshooter))
    :effect ((held-up robbie bank)))
```

Figure 3.39: The 'hold-up-bank' operator instantiated with the bindings (robbie sixshooter bank).

Plan. Since the story does not contain a frame of commitment with intentions matching any of the effects of HOLD-UP-BANK-7, the planner chooses the frame of commitment with the intention (intends robbie (has robbie big-money)) and inserts this frame into the Discourse Plan as FRAME-1. Open preconditions are created for each of the preconditions of HOLD-UP-BANK-7 and a new link to intention flaw is created with HOLD-UP-BANK-7 and FRAME-1. The resulting plan is displayed in Figure 3.40.

The open precondition flaws are addressed in order. The preconditions (person robbie) and (evil robbie) are linked to HATCH-PLAN-2, already in the Discourse Plan, and the precondition (bank bank) is linked to WITHDRAW-MONEY-4. Because none of the steps in the Discourse Plan establish (gun sixshooter) a new story step (HOLSTER-GUN robbie sixshooter)- indicating that Robbie puts the Six Shooter in his holster - is selected with the desired precondition and instantiated as HOLSTER-GUN-8. HOLSTER-GUN-8 also serves to establish (has robbie sixshooter) and is linked to this precondition. The last precondition is (at robbie bank). A new story step (RIDE-HORSE-TO robbie brownhorse mainstreet bank) - indicating that Robbie rides the Brown Horse from Main Street to the Bank - is instantiated as RIDE-HORSE-TO-9 and linked to this precondition. The resulting plan is displayed in Figure 3.41.

The planner addresses the link to intention flaw with HOLD-UP-BANK-7 and FRAME-1 next. Here either the routine in Figure 3.24 or the routine in Figure 3.25 might be employed. However, none of the effects of HOLD-UP-BANK-7 fulfill the intention of FRAME-1, (has robbie big-money), and the routine in Figure 3.25 must be selected to insert a new inferred step into the Discourse Plan. The COLLECT-MONEY-FROM-HEIST

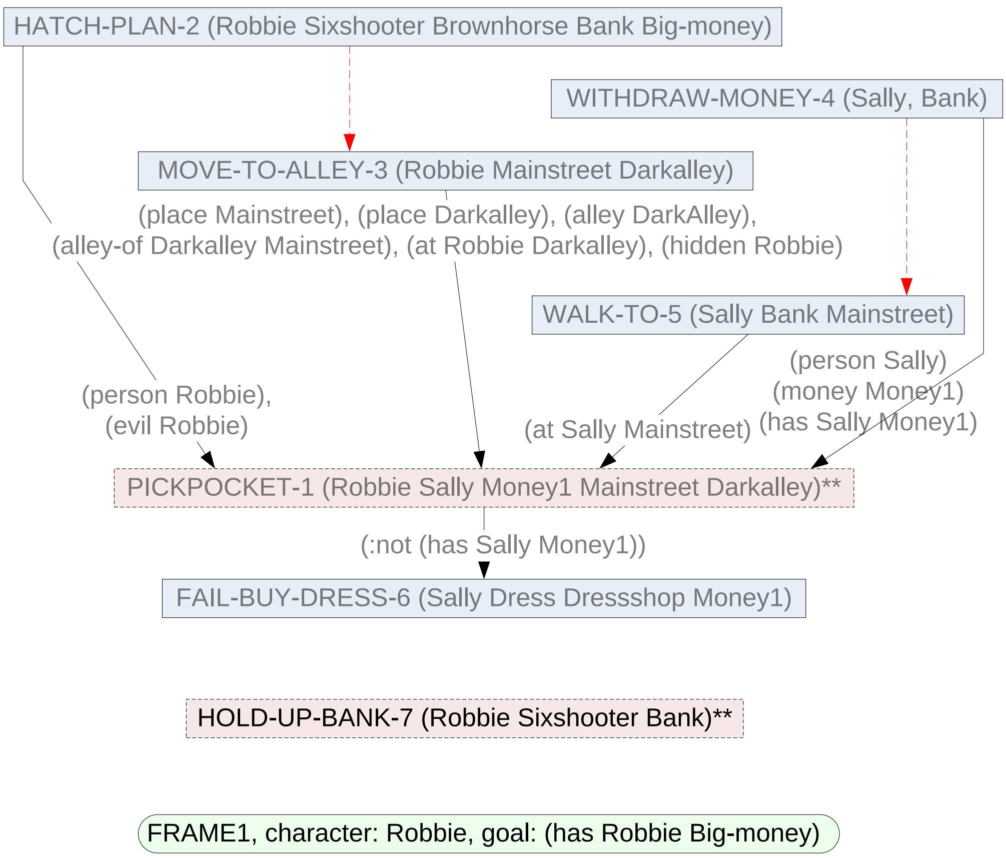

Figure 3.40: The Discourse Plan after resolving the intentional inference flaw for HOLD-UP-BANK. Earlier additions are lightly shaded in this figure. Current flaws: *Open Precondition:* (person robbie) (evil robbie) (bank bank) (gun sixshooter) (at robbie bank) (has robbie sixshooter). *Link to Intention:* (HOLD-UP-BANK-7 FRAME-1). *Focus:* SIXSHOOTER.

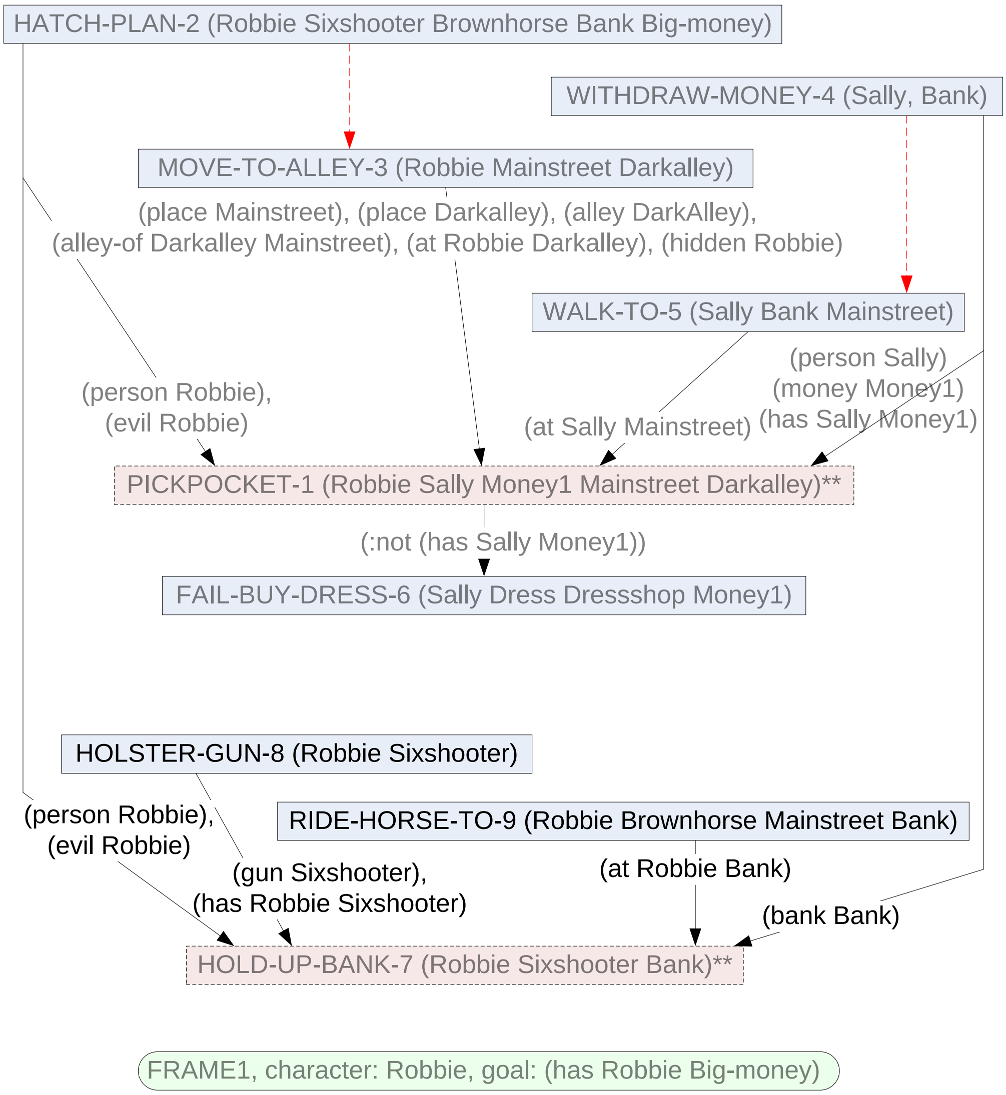

Figure 3.41: The Discourse Plan after resolving open precondition flaws for HOLD-UP-BANK-7. Earlier additions are lightly shaded in this figure. Current flaws: *Link to Intention:* (HOLD-UP-BANK-7 FRAME-1). *Focus:* SIXSHOOTER.

```
(define (action collect-money-from-heist)
    :parameters (?person ?bank ?mother-lode)
    :actors (?person)
    :precondition ((person ?person) (evil ?person) (bank ?bank)
                   (mother-lode ?mother-lode) (at ?person ?bank)
                   (held-up ?person ?bank) (has ?bank ?mother-lode))
    :effect ((has ?person ?mother-lode)
             (:not (held-up ?person ?bank))
             (:not (has ?bank ?mother-lode))))
```

Figure 3.42: The collect-money-from-heist' operator.

```
(define (action collect-money-from-heist-10)
    :parameters (robbie bank big-money)
    :actors (robbie)
    :precondition  ((person robbie) (evil robbie) (bank bank)
                    (mother-lode big-money) (at robbie bank)
                    (held-up robbie bank) (has bank big-money))
    :effect ((has robbie big-money)
             (:not (held-up robbie bank))
             (:not (has bank big-money))))
```

Figure 3.43: The collect-money-from-heist' operator instantiated with the bindings (robbie bank big-money).

operator in Figure 3.42 is chosen and instantiated as COLLECT-MONEY-FROM-HEIST-10 with the bindings shown in Figure 3.43. The effect of (has robbie big-money) is linked to FRAME1, satisfying the intention. New open precondition flaws are created for the preconditions of COLLECT-MONEY-FROM-HEIST-10 and a new link to intention flaw is created for HOLD-UP-BANK-7 and COLLECT-MONEY-FROM-HEIST-10. The resulting plan is displayed in Figure 3.44.

The planner addresses the new link to intention flaw, this time employing the routine in Figure 3.24 to link the two steps directly and complete the causal chain to FRAME1. A causal link is created from HOLD-UP-BANK-7 to COLLECT-MONEY-FROM-HEIST-10 over the precondition (held-up robbie bank). The open precondition flaws for COLLECT-MONEY-FROM-HEIST-10 are addressed next. The preconditions (person robbie), (evil robbie), (bank bank), and (at robbie bank) are linked to earlier establishing steps. A new step (DELIVER-MONEY steve big-money bank) - indicating that Steve delivered the Big Money to the Bank - is instantiated as DELIVER-MONEY-11 and used to fulfill the preconditions (mother-lode big-money) and (has bank big-money). The resulting plan is displayed in Figure 3.45. This completes the refinement of all the flaws relating to the intentional inference flaw; the plan now contains enough information to intentionally necessitate and enable the inference.

The last remaining flaw is a focus flaw denoting that the 'SIXSHOOTER' object receive a total focus of at least 1.0. The routine in Figure 3.26 is used to refine the flaw. The MEI model is employed to total the focus for this object in the current Discourse Plan, and the result is 0.8. To increase the focus of SIXSHOOTER, the story step (TAKE-THING-OFF-SLEEPER ROBBIE BARNEY SIXSHOOTER DARKALLEY) - indicating that Robbie took the Six Shooter off of Barney sleeping in the Dark Alley - is instantiated as TAKE-THING-OFF-SLEEPER-12 and added to the Discourse Plan. A causal link is transferred from the story leading from TAKE-THING-OFF-SLEEPER-12 to HOLSTER-GUN-8 over (has robbie sixshooter), and a number of orderings are added to constrain the position of TAKE-THING-OFF-SLEEPER-12. The focus is recomputed for SIXSHOOTER, and the new value is above the threshold. No flaws remain in the Discourse Plan; a solution has been found. The final Discourse Plan is displayed in Figure 3.46, a linearization is given in Figure 3.47, and Figure 3.48 is a text translation of the linearization.

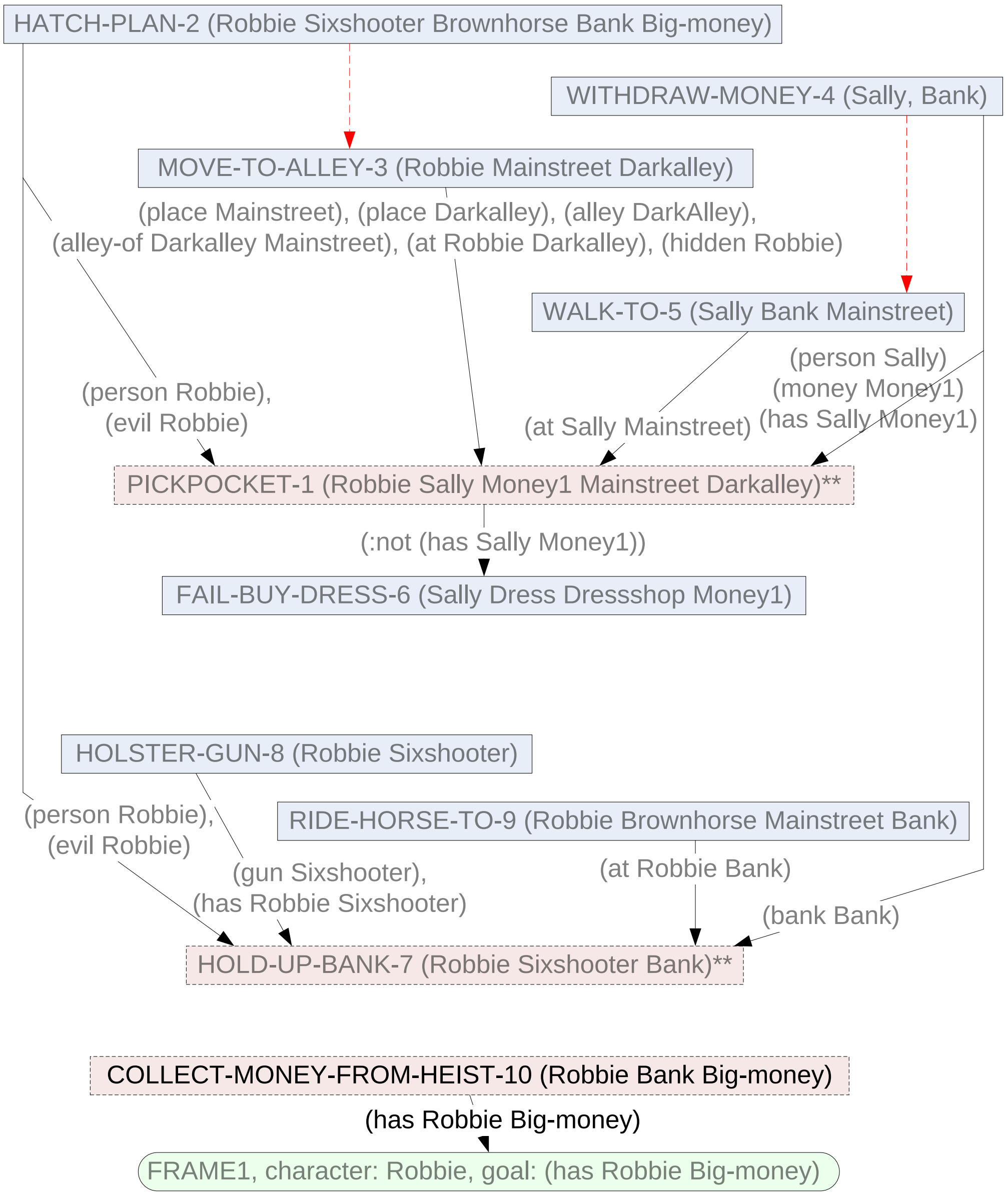

Figure 3.44: The Discourse Plan after resolving the link to intention flaw for HOLD-UP-BANK-7 and FRAME-1. Earlier additions are lightly shaded in this figure. Current flaws: *Open Precondition:* (person robbie) (evil robbie) (bank bank) (mother-lode big-money) (at robbie bank) (held-up robbie bank) (has bank big-money). *Link to Intention:* (HOLD-UP-BANK-7 COLLECT-MONEY-FROM-HEIST-10). *Focus:* SIXSHOOTER.

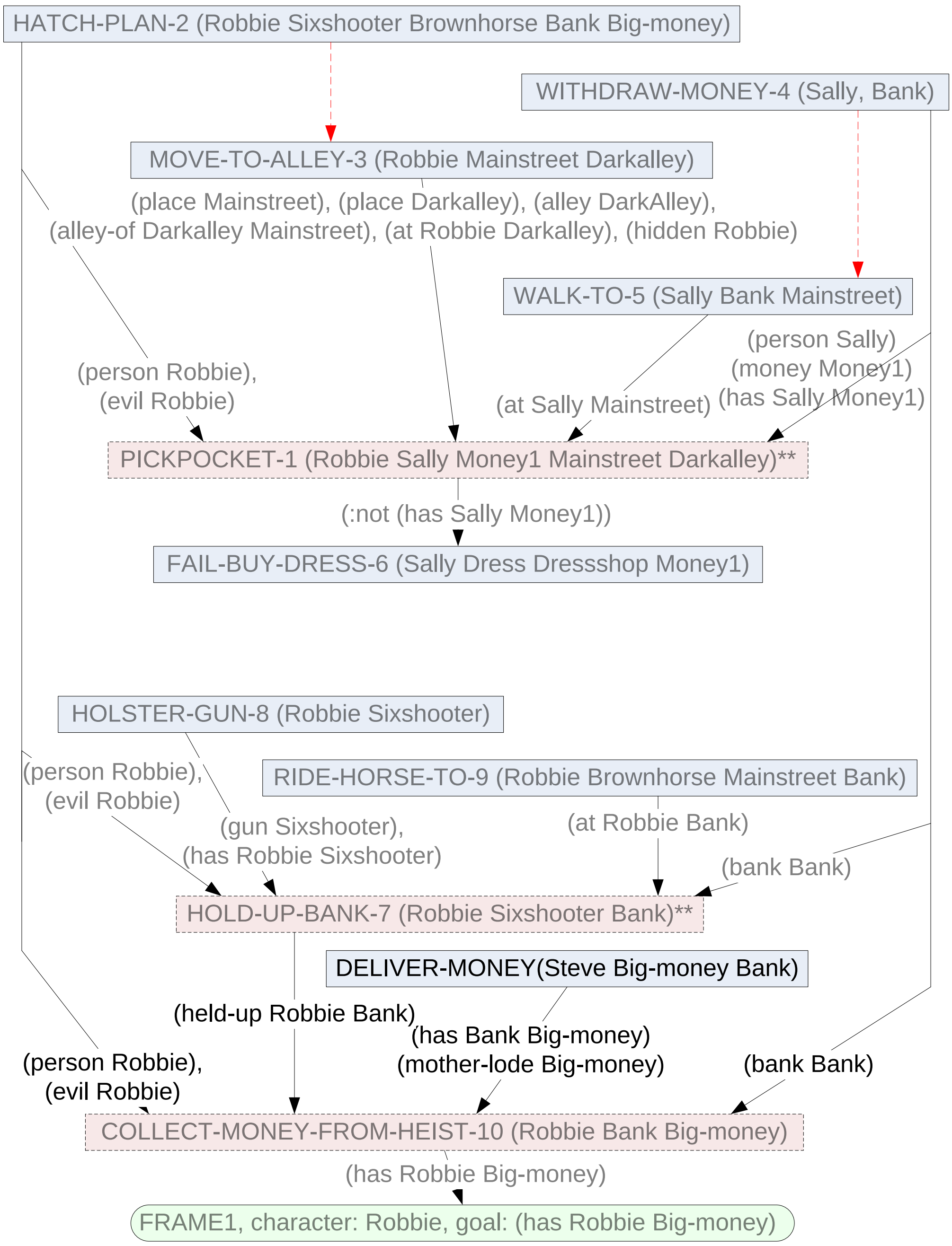

Figure 3.45: The Discourse Plan after resolving the link to intention flaw for HOLD-UP-BANK-7 and COLLECT-MONEY-FROM-HEIST-10 and the open precondition flaws for COLLECT-MONEY-FROM-HEIST-10. Earlier additions are lightly shaded in this figure. Current flaws: *Focus:* SIXSHOOTER.

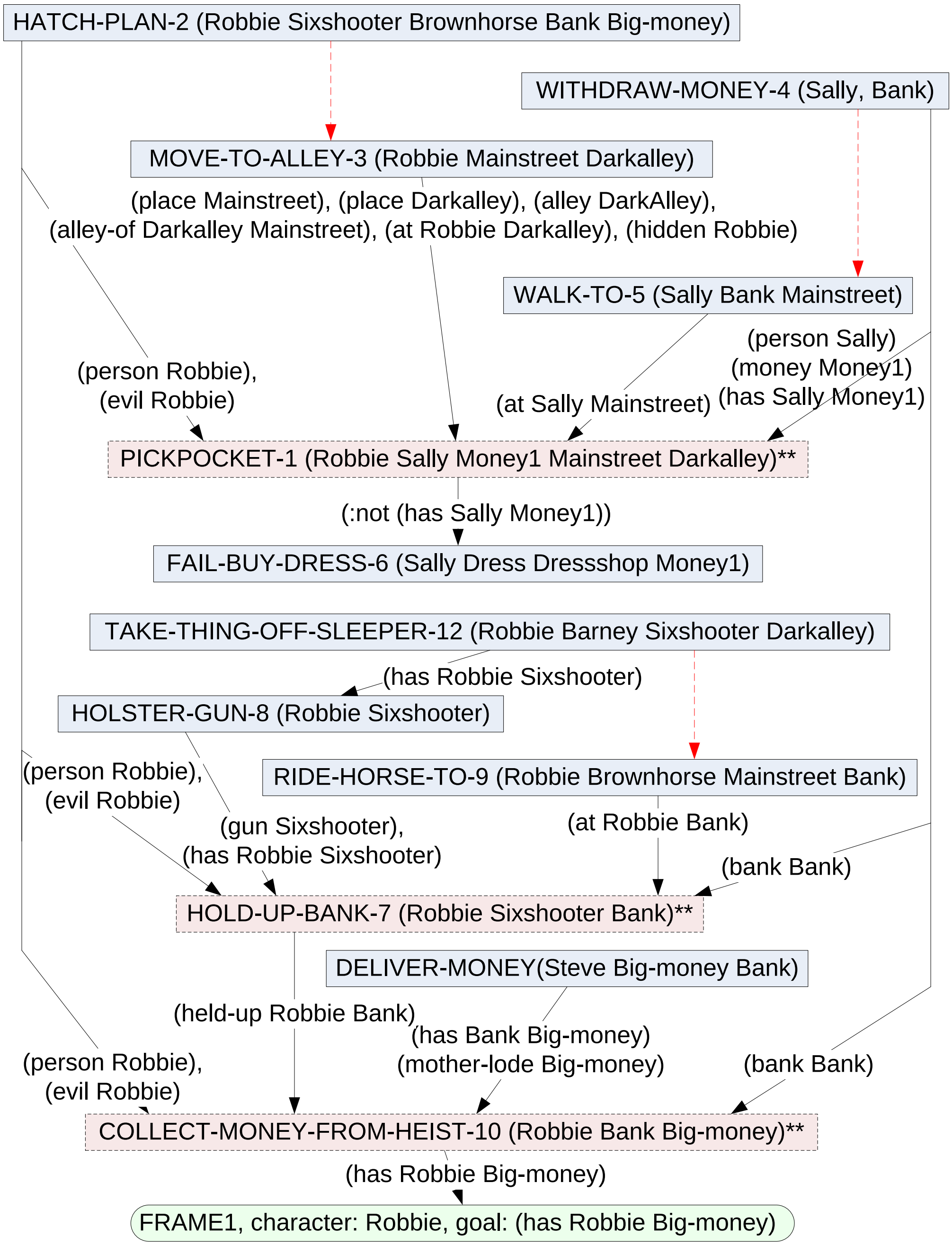

Figure 3.46: The solution Discourse Plan. TAKE-THING-OFF-SLEEPER-12 has been added to resolve the focus flaw.

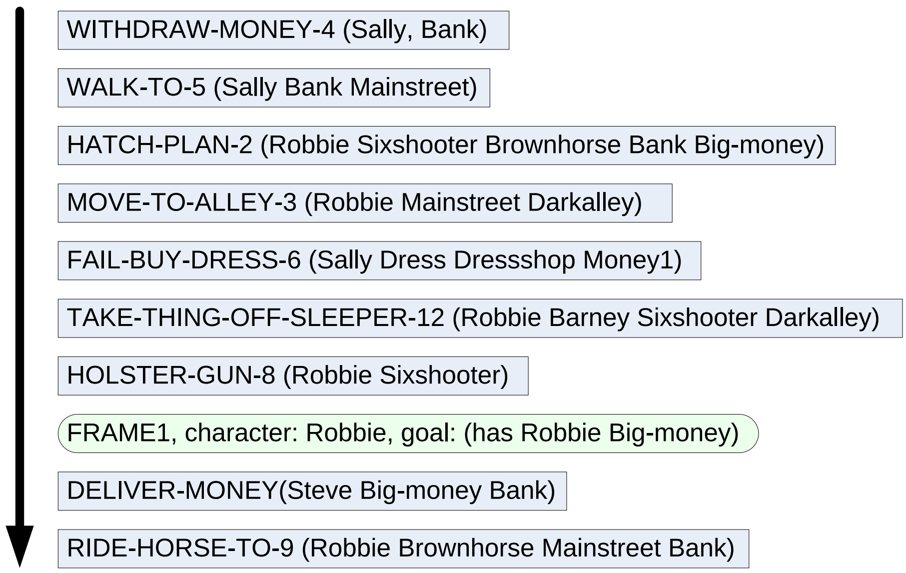

Figure 3.47: The solution Sequence of Discourse Content, one of the linearizations of the Discourse Plan in Figure 3.46.

1. Sally withdrew some money from the Town Bank.
2. She walked out of the Bank onto Main Street.
3. Robbie hatched an evil plan.
4. Robbie hid in the dark alley off of Main Street.
5. Sally reached in her purse to pay for her new dress, but her money was gone!
6. Robbie took the six shooter of the guard sleeping in the alley.
7. He holstered the six shooter.
8. Robbie wanted gold, lots of gold.
9. Steve delivered the gold bars to the Town Bank.
10. Robbie rode his horse down Main Street to the Bank.

Figure 3.48: Text translation of the Sequence of Discourse Content in Figure 3.47.

Chapter 4

Experimental Evaluation

This chapter presents an evaluation of INFER by three separate experiments. The first experiment tests aspects narrative focus, observing reading under conditions of varying levels of narrative focus as predicted by the MEI model. The second experiment tests the inference prediction of the MEI model over hand-encoded narratives, observing word recognition time under prompted and non-prompted inference conditions. The third experiment tests the ability of INFER to generate Sequences of Discourse Content to prompt inferences. The results of these experiments provide support for the computational model of narrative comprehension and generation presented in this work.

The first experiment was designed to test aspects of the MEI model of narrative focus. Sentence and word reading times have been shown to relate to measures of focus and activation [11] [9]. The more activated the elements in the sentence are, the easier and faster readers can read them. Because MEI calculates the *relatedness* of the sentence as the average activation of its elements before integration, the hypothesis for experiment 1 is that sentences which have higher relatedness will be read faster and have shorter average reading times. To test this hypothesis, 4 experimental narratives were constructed to have sentences with a range of relatedness. Participants read the narratives one line at a time, and reading times were recorded and regressed against the relatedness rating. In addition to the regression, two conditions of each story were constructed; high and low relatedness. In the high relatedness condition a test sentence was placed in the narrative such that it had a high relatedness score, and in the low relatedness condition the same test sentence was relocated in the where such that it had a low relatedness score. The experiment is described in more detail in Section 4.1. The method is given in Section 4.1.1; the results of this experiment are presented in Section 4.1.2 and discussed in Section 4.1.3.

The second experiment was designed to test aspects of the MEI model of narrative inferencing. Word recognition times have been used by psychological studies of narrative comprehension to measure the construction of inferences [73]. Inferences related to elements in the discourse increase the availability of those elements in recognition and recall tests. If a reader infers that a character will use a paint brush mentioned in the narrative to paint a picture, then the reader will recognized the word brush as being in the story faster than if the inference was not made. The MEI model predicts inferences from a give Sequence of Discourse Content. The hypothesis for experiment 2, then, is that readers will recognize words related to inferences faster. To test this hypothesis, 8 experimental narratives were constructed, 4 using causal necessitation and 4 using intentional necessitation. Each narrative had two conditions, one in which the inference was necessitated and enabled and one in which it was not. Participants read the narratives one line at a time, and then answered word recognition tests immediately following each narrative - word recognition times were recorded. The experiment is described in more detail in Section 4.2. The method is given in Section 4.2.1; the results of this experiment are presented in Section 4.2.2 and discussed in Section 4.2.3.

The third experiment was designed to test the ability of INFER to generate Sequences of Discourse Content to prompt inferences. Since this experiment is again testing the construction of inferences, the same word recognition time measurements as experiment 2 were employed. An extended event log was developed to use as the story input to INFER. The event log was similar to that used in the example in Section 3.7, detailing the events in a western town leading up to a bank robbery. This event log was used as the story input to INFER, and Sequences of Discourse Content were generated using single causal or intentional inference flaws, creating sequences prompting a single inference. These sequences were translated to text using simple templates, and used as the prompted condition for the experiment. The unprompted condition was constructed by switching one or more events from the sequence until INFER no longer predicted the inference. Participants read the narratives one line at a time, and then answered word recognition tests immediately following each narrative - word recognition times were recorded. The experiment is described in more detail in Section 4.3. The method is given in Section 4.3.1; the results of this experiment are presented in Section 4.3.2 and discussed in Section 4.3.3.

4.1 Experiment 1

This experiment tests aspects of the MEI model of narrative focus. Sentence and word reading times have been shown to relate to measures of focus and activation [11] [9]. The more activated the elements in the sentence are, the easier and faster readers can read them. Because MEI calculates the *relatedness* of the sentence as the total activation of its elements before integration, the hypothesis for experiment 1 is that sentences which have higher relatedness will have shorter average reading times. To test this hypothesis, narratives were created to include sentences with various levels of relatedness, from very low relatedness to very high relatedness, and sentence reading times were recorded and regressed against the relatedness rating.

The full texts of the narratives used for this experiment and the instructions given to participants are presented in Appendix 5.4.

4.1.1 Method

Materials. The narratives for the first experiment consisted of 4 experimental narratives and 8 filler narratives. The 4 experimental narratives were created by creating a complete and sensical IPOCL story plan, selecting a Sequence of Discourse Content from the story plan, and generating text for the sequence using simple text templates. The resulting text was used for the experiment.

In addition, each of the 4 experimental narratives had two conditions; high relatedness and low relatedness. In the high relatedness condition, a test sentence was placed in the narrative such that it had a high relatedness score. In the low relatedness condition, this test sentence was swapped with a sentence that had a low relatedness score in that context. The texts of these narratives are presented in Appendix 5.4.

Procedure. The narratives were displayed one line at a time by a Java applet within a browser window. The experiment began with instructions to read for comprehension and a short example narrative. The participants were not informed they would be timed. Participants placed their hand on a computer mouse. The participants advanced the narrative one sentence at a time by pressing the left mouse button. After each mouse click, the current sentence on screen was erased, and the next sentence was displayed immediately. Participants could not return to earlier sentences. Reading times were recorded after each sentence.

Table 4.1: Reading Time Summary Statistics, Experiment 1

N	Mean	Median	Std Dev	Minimum	Maximum	Skewness
745	2,583ms	2,186ms	1,605ms	708ms	14,215ms	2.823

At the end of each narrative two word recognition tests were administered. The participant was presented with the question as to whether a particular word appeared in the narrative and indicated either 'yes' or 'no' by selecting an option with the mouse. There was a 500-ms pause between questions and between a question and the start screen for the next narrative. Participants were given untimed rest periods after the first two sets of 4 narratives, indicated by onscreen instructions. The instructions for this experiment are presented in Appendix 5.4.

Design and Subjects. 22 middle school teachers participating in a technology research program volunteered to participate. Each participant only read one condition of each narrative. The narrative conditions were counter-balanced in a rotating sequence, and the order of narratives given to each participant was randomized.

4.1.2 Results

Table 4.1 presents the summary statistics for reading times across all sentences and all participants. The average reading time was about 2.5 seconds with a large standard deviation at 2.1 seconds. The skewness of 2.8 indicates that there is a long tail to the right. These statistics are typical of reading time measurements.

A Bivariate regression analysis carried out to model how reading times vary with relatedness ratings found that relatedness was a small, but statistically significant predictor of reading times ($\beta = -0.175$, $a = 2.98$), confirming the hypothesis. Sentence reading times dropped an average of 175 milliseconds with for each point increase in relatedness score. The relationship was statistically significant: $F(1, 743) = 32.28$, $p < 0.0001$, r^2 $= 0.0416$, indicating that about 4% of the variance in reading time was explained by the relatedness score. Figure 4.1 displays the scatter plot and regression line. The relationship held when reading times were scaled by the number of words per sentence $F(1, 743) = 10.82$, $p < 0.0011$. Reading times for initial, introductory sentences (e.g. "There was a boy named Billy.") were not used in the analysis since their relatedness was 0. This removed one to four sentences from the beginning of each narrative.

A repeated measures ANOVA was employed to test the difference in the reading

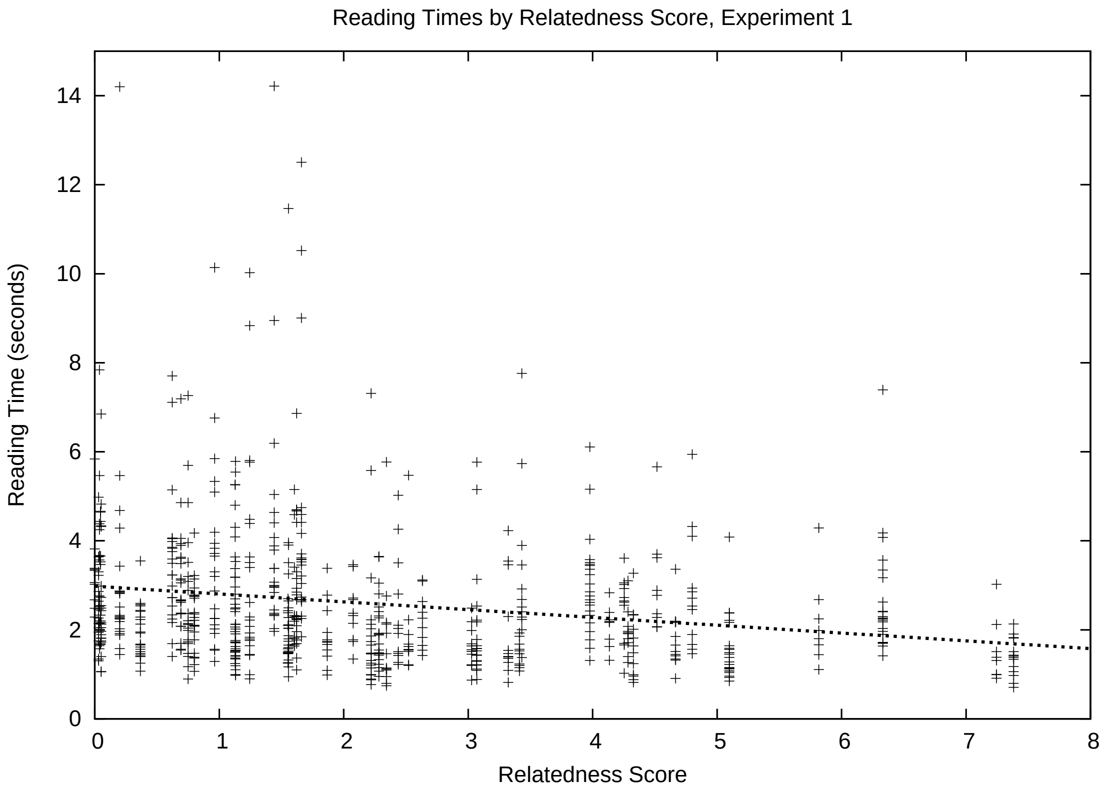

Figure 4.1: Scatter plot of reading times vs. relatedness scores with regression line, $p < 0.0001$. Sentences are read faster when they are more related.

Table 4.2: Reading Time Summary Statistics between Conditions, Experiment 1

condition	N	Mean	Median	Std Dev	Minimum	Maximum	Skewness
high	43	2,454ms*	1,433ms	2,168ms	885ms	7,761ms	1.944
low	44	3,000ms*	1,292ms	2,677ms	1,344ms	7,842ms	1.658

$p < .05$*

times of key sentences between the high relatedness and low relatedness conditions. The ANOVA showed a significant effect $F(1, 86) = 4.78$, $p < 0.05$ indicating that reading times were slower for the low relatedness condition, confirming the hypothesis. Table 4.2 presents the summary statistics for each condition. The difference in conditions held when reading times were scaled by the number of words per sentence $F(1, 86) = 5.36$, $p < 0.05$.

4.1.3 Discussion

The results of this experiment support the MEI model of focus in narrative comprehension. The relatedness score averaged the activation of the elements in the sentence to obtain an overall activation for the sentence. As predicted by the hypothesis, sentences with higher relatedness had shorter average reading times. The regression analysis shows this general trend for non-test the sentences in the narratives, reading times decreased an average of 175ms for each additional point on the relatedness score. The repeated measures ANOVA shows this effect in a small set of test sentences. When the test sentences were highly related they were read faster than in the low relatedness condition. These results show that the MEI model can accurately predict narrative focus comprehension effects in readers.

4.2 Experiment 2

The second experiment was designed to assess the availability of inference related information at the end of reading short narratives. The design of this study is similar to studies in the discourse processing literature that also test models of inference generation [27] [73]. This study tests online inferences without reading strategy instruction. These are inferences that are made during reading without specific instruction to do so, sometimes termed automatic inferences. If the inferences are constructed by the reader, the inferences will activate related information in the reader's mental model of the story and make this information more available at the end of the narrative.

The 8 experimental narratives consisted 4 narratives designed to test causally necessitated inferences and 4 narratives designed to test intentionally necessitated inferences. In addition, 4 filler narratives were included. The narratives were created by first creating an IPOCL story plan, selecting a Sequence of Discourse Content from the plan, and then using text templates to translate the sequence to text.

Each of the test narratives had two conditions: the prompted condition necessitated and enabled the appropriate inference according to the MEI model, and the unprompted condition did not predict and inference due to either lack of necessitation or enablement. Participants read through the narratives one sentence at a time in a self paced manner. Reading times were recorded for each sentence. After each narrative, participants were asked to answer two word recognition questions, stating whether a given word was in the narrative or not. One of these words related to the prompted inference. Response times and error rates were recorded.

The overall hypothesis was that the experimental condition would change the word recognition response times. In the causally necessitated inference narratives, readers were asked about a word not in the text but was the action prompted in inference. If the inference was that “Jimmy bought the guitar.” then the test word was ’bought’. The hypothesis for these narratives was that participants would take longer to answer this test in the prompted condition, because they would made the inference and would be attempting to decide if the word was actually in the text. In the intentionally necessitated inference narratives, readers were asked about a word in the stated intention of the character. If the character intention was “Draco wanted to cook and eat Olive.”, ’cook’ might be the test word. The hypothesis for these narratives was that participants would take less time to answer this test in the prompted condition, having read the word and made related inferences.

The full texts of the narratives used for this experiment and the instructions given to participants are presented in Appendix 5.4.

4.2.1 Method

Materials. The narratives consisted of 4 filler narratives and 8 experimental narratives divided between 4 narratives designed to test causally necessitated inferences and 4 narratives designed to test intentionally necessitated inferences. The narratives were created by first creating an IPOCL story plan, selecting a Sequence of Discourse Content from the plan, and then using text templates to translate the sequence to text.

Each of the test narratives had two conditions: the prompted condition necessitated and enabled the appropriate inference according to the MEI model, and the unprompted condition did not predict and inference due to either lack of necessitation or enablement. The conditions were the same length in sentences and approximately the same number of words per sentence. The narratives ranged from 9 to 21 sentences each, and at most 4 lines were changed between conditions. The full texts of these narratives are presented in Appendix 5.4.

Procedure. The narratives were displayed one line at a time by a Java applet within a browser window. The experiment began with instructions and a short example narrative. Participants were asked to read for comprehension, and were not informed they would be timed. Participants placed their hands on a standard QWERTY keyboard so that their left index finger was on the 'z' key, their right index finger was on the '/' key and their thumbs were on the space bar. The participants advanced the narrative one sentence at a time by pressing the spacebar. After each press of the spacebar, the current sentence on screen was erased, there was a 200-ms pause, and the next sentence was displayed. Participants could not return to earlier sentences.

At the end of each narrative two word recognition tests were administered. The participant was presented with the question as to whether a particular word appeared in the narrative and indicated either 'yes' by pressing the 'z' key or 'no' by pressing the '/' key. There was a 500-ms pause between questions and between a question and the start screen for the next narrative. Participants were given untimed rest periods after the first two sets of 4 narratives, indicated by onscreen instructions. The instructions for this experiment are presented in Appendix 5.4.

Design and Subjects. 18 students from an introductory Artificial Intelligence class participated in the study for course credit. Each participant only read one condition of each narrative. The narrative conditions were counter-balanced in a rotating sequence, and the order of the narratives given to each participant was randomized.

Table 4.3: Intentional Narrative Results

	RT mean	RT StdDev	% error	rd mean
prompted	3,304 ms	2,148 ms	11.1	394 ms
unprompted	3,179 ms	2,126 ms	9.7	401 ms

Note RT = response time, rd = reading time

Table 4.4: Causal Narrative Results

	RT mean	RT StdDev	% error	rd mean
prompted	4,384 ms	3,149 ms	6.9	596 ms
unprompted	4,054 ms	3,544 ms	4.2	591 ms

Note RT = response time, rd = reading time

Table 4.5: Intentional Narratives, Slow Readers

	RT mean	RT StdDev
prompted	4,484 ms	2,901 ms
unprompted	5,763 ms	4,324 ms

Table 4.6: Intentional Narratives, Fast Readers

	RT mean	RT StdDev
prompted	4,285* ms	3,462 ms
unprompted	2,346* ms	990 ms

* $p < .069, F(1, 18) = 3.53$

Table 4.7: Response Time Summary Statistics between Conditions (Intentional), Experiment 2

condition	N	Mean	Median	Std Dev	Minimum	Maximum	Skewness
prompted	36	4,384ms	3,472ms	3,149ms	1,027ms	12,834ms	1.171
un-prompted	36	4,054ms	2,792ms	3,544ms	932ms	15,441ms	1.904

4.2.2 Results

The overall results are presented in tables 4.3, 4.4, 4.7. Reading times per word are included for completeness. We did not form any hypotheses for the reading times.

In the intentional narratives, participants were slightly faster to respond in the unprompted condition, and the error rate was slightly less in this condition as well. They were able to recall that a word relating to the intention was part of a narrative slightly faster and more accurately when the events in the narrative did not heavily involve that intention or allow for inferencing involving that intention. This result is counter to our original hypothesis.

In the causal narratives, participants were slightly faster to respond in the unprompted condition, and the error rate was slightly less. They were able to recall correctly that a word was *not* in the narrative slight faster and more accurately when the narrative did not prompt an inference related to the word. The prompting of the inference appears to slow down response time and increase error rate. This result is in line with our original hypothesis (though without statistical significance).

Two interesting trends emerge from the data. The first is that reading times correlate significantly with response times, $R^2 = .60, p < .001$. The second is that the variances for the response times are quite large; one standard deviation away from the mean ranges from 1 to 5 seconds in the intentional narratives. Using these observations, participants were split into two groups: slow readers (above median) and fast readers(equal or below median) based on their average reading time. The effect of the conditions on these two groups were analyzed.

Tables 4.5 and 4.6 show the results of the second analysis for the intentional narratives. Figure 4.2 is the box plot of the log response times. In the slow readers group, response time was faster in the prompted condition. Adding actions related to the intention seemed to improve availability, as per our hypothesis. In the fast readers group, prompting slowed the response time dramatically. Adding actions related to the intention seemed to hinder availability. This last result approached significance at $p < .069$.

4.2.3 Discussion

The experimental condition changed overall response times, providing support for the general hypothesis. In the intentionally necessitated inference narratives, the effect of

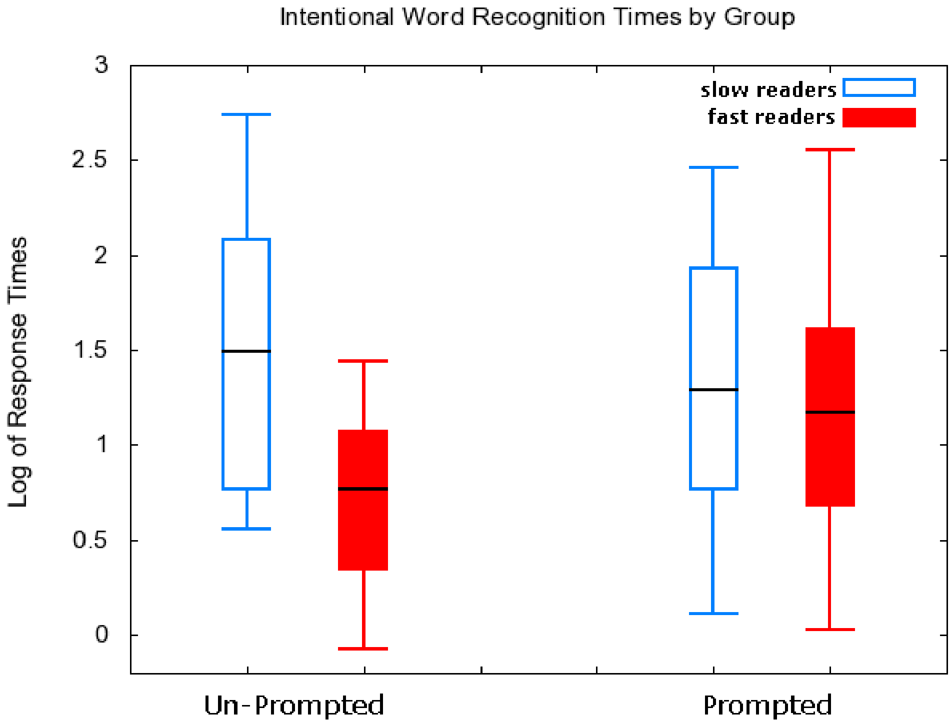

Figure 4.2: Box plot of Experiment 2 results.

the inferences depended on whether the participants were slow or fast readers. In the slow reader group, the addition of the inferences seems to make the information more available to the reader. In this case, the reader may be using the semantics of the narrative to recall the information, and is thus aided by the inferences. In the fast reader group, the prompting of intentional inferences significantly slows the response time. In this case, the participant may be relying on the surface text of the narrative more than the semantics. Reading time is faster because the reader does not have to encode the semantics of the narrative, but response time is hindered by related information. The participant takes longer to recall a specific item out of a collection of highly related items than to recall a specific item out of a collection of unrelated items.

In the causally necessitated inference narratives, the results provide weak support for the specific hypothesis. The addition of causal inferences relating to a word not in the text seem to slow response times and increase error rates. The information may be more salient in the reader's mind and the reader may have a more difficult time discerning whether the information was inferenced or included.

The difference in response times over the prompted conditions supports the claim that the MEI model is able to predict online, story-level inferences. However, these results may be improved by a slightly modified experiment. Participants may have optimized their reading strategy to address the word recognition questions after each narrative. Including a comprehension question before the word recognition questions may result in more comprehension based reading. In addition, response times were highly variable, and perhaps the switch from reading to responding was interfering with the experimental measure. Including more word recognition questions for each narrative may reduce variability in the later questions. These recommendations are incorporated in experiment 3.

4.3 Experiment 3

This experiment was designed to assess the ability of INFER to generate Sequences of Discourse Content that prompt inferences. The design of this study is similar to studies in the discourse processing literature which also test models of inference generation [27] [73]. This study tests online inferences without reading strategy instruction. These are inferences that are made during reading without specific instruction to do so, sometimes termed automatic inferences. If the inferences are constructed by the reader, the inferences will activate related information in the reader's mental model of the story and make this information more available at the end of the narrative.

Four experimental narratives were generated with inferencing criteria from a sample event log using the method described in Chapter 3. An extended event log was developed to use as input to INFER. The event log was similar to that used in the example in Section 3.7, detailing the events in a western town leading up to a bank robbery. This event log was used as the story input to INFER, and Sequences of Discourse Content were generated using single causal or intentional inference flaws, creating sequences prompting a single inference. These sequences were translated to text using simple templates, and used as the prompted condition for the experiment. Two of the four experimental narratives employed causally necessitated inferences and the other two employed intentionally necessitated inferences. The unprompted condition was constructed by switching one or more events from the sequence with steps from the story until INFER no longer predicted the inference. In addition 8 filler narratives were included in the experiment. Participants read through the narratives one sentence at a time in a self paced manner. Reading times were recorded.

After each narrative, participants were asked to answer one comprehension question and then 5 word recognition questions, pressing a key to indicate whether a given word was in the narrative or not. One of these words related to the prompted inference, while appearing in the story (for both causally and intentionally necessitated inferences). The hypothesis for this experiment that in the prompted condition readers would take less time to answer this test than in the unprompted condition.

The full texts of the narratives used for this experiment and the instructions given to participants are presented in Appendix 5.4.

4.3.1 Method

Materials. The narratives consisted of 8 filler narratives and 4 experimental divided between 2 narratives designed to test causally necessitated inferences and 2 narratives designed to test intentionally necessitated inferences. An extended event log was developed to use as input to INFER. The event log was similar to that used in the example in Section 3.7, detailing the events in a western town leading up to a bank robbery. Events included purchasing a dress, a pickpocketing, drinking at the saloon, and a bank robbery. This event log was used as the story input to INFER, and Sequences of Discourse Content were generated using single causal or intentional inference flaws, creating sequences prompting a single inference. These sequences were translated to text using simple templates, and used as the prompted condition for the experiment. Two of the four experimental narratives employed causally necessitated inferences and the other two employed intentionally necessitated inferences. The unprompted condition was constructed by switching one or more events from the sequence with steps from the story until INFER no longer predicted the inference. 2 narratives were generated each with a single causal inference goal, and 2 narratives were generated each with a single intentional inference goal. A simple templating function was used to map the narrative elements into a list of human readable sentences. The narratives ranged between 5 and 6 sentences each, and 1 line was changed between conditions. There was one short answer comprehension question and 5 word recognition questions after each narrative.The full texts of the narratives and questions are presented in Appendix 5.4.

Procedure. The narratives were displayed by a Java applet within a browser window. The experiment began with instructions and a short example narrative. In this experiment, the participants were instructed to read for comprehension and were informed they would be timed. Participants placed their hands on a standard QWERTY keyboard so that their left index finger was on the 'z' key, their right index finger was on the '/' key and their thumbs were on the space bar. The participants advanced the narrative one sentence at a time by pressing the spacebar. After each press of the spacebar, the current sentence on screen was erased, there was a 200-ms pause, and the next sentence was displayed. Participants could not return to earlier sentences.

At the end of each narrative one comprehension question and five word recognition tests were administered. The comprehension question was a short answer question about a step in the narrative. The participants answered by typing on the keyboard. Next, participants were presented with the question as to whether a particular word appeared in the narrative and indicated either 'yes' by pressing the 'z' key or 'no' by pressing the '/' key. There was a 500-ms pause between questions and between a question and the start screen for the next narrative. Participants were given untimed rest periods after the first two sets of 4 narratives, indicated by onscreen instructions. The instructions for this experiment are presented in Appendix 5.4.

Design and Subjects. 40 students from an undergraduate computer science course and 8 students from a graduate computer science course participated in the study for course credit. Each participant only read one condition of each narrative. The narrative conditions

Table 4.8: Intentional Narrative Results

	RT mean	RT StdDev	% error	rd mean
prompted	957* ms	376 ms	6.1	2,703 ms
unprompted	1,246* ms	854 ms	4.1	2,818 ms

* $p < .05$ *Note* RT = response time, rd = reading time

Table 4.9: Causal Narrative Results

	RT mean	RT StdDev	% error	rd mean
prompted	1,097 ms	493 ms	10.2	3,333ms
unprompted	1,171 ms	528 ms	12.2	3,061ms

Note RT = response time, rd = reading time

were counter-balanced in a rotating sequence, and the order of the narratives given to a participant was randomized.

4.3.2 Results

The results of our study are presented in tables 4.8 and 4.9. Figure 4.3 is the box plot of recognition times for the intentionally necessitated inference narratives. Table 4.8 shows the mean response times for the critical word recognition tests in the intentional inferencing narratives, and Table 4.9 shows the mean response times for the critical word recognition tests in the causal inferencing narratives. Reading times of the last sentence are included for completeness. We did not form any a priori hypotheses for the reading times.

For the intentionally necessitated inference narratives, a two factor mixed model ANOVA on the response times indicated a significant effect for prompted condition $F(1, 73.4) = 4.39, p < .05$, but no effect for narrative $F(1, 47.2) = 2.34$ and no interaction $F(1, 80.9) = .02$. The average response time was shorter when the word was prompted by an inference condition. This result indicates that the intentional inferences were constructed online and that they had an effect on reading comprehension.

For the causally necessitated inference narratives, a two factor mixed model ANOVA on the response times did not indicate a significant effect for prompted condition $F(1, 76.2) = 0.93$, but narrative was significant at $F(1, 47.4) = 17.12$ with no interaction $F(1, 78.3) = 1.15$.

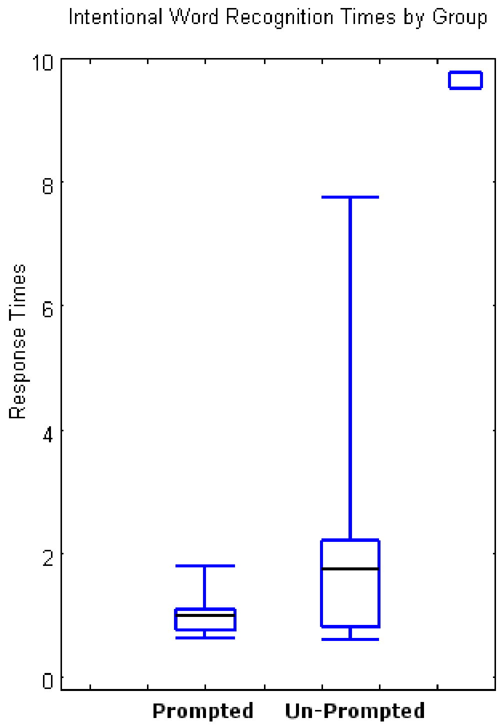

Figure 4.3: Box plot of Experiment 3 results. Reading times in seconds.

4.3.3 Discussion

These results provide strong support for INFER's generation of Sequences of Discourse Content to prompt inferences. Sequences generated to prompt these inferences had a significant effect on the comprehension measure of word recognition times. The intentionally necessitated inferences had a strong effect on recognition times. Recognition times decreased nearly 25% when in conjunction with a prompted inference.

The causal inferencing narratives, while decreasing response time, did not show a significant effect due to prompting. This may be because intentionally necessitated inferences are forward looking, often persisting for multiple steps in the sequence, while causal inferences are backward looking, often ending after a single step. Thus, causally necessitated inferences are enabled for shorter periods of time. For this reason their effect on comprehension may be less, making them more difficult to detect.

4.4 Discussion

The collection of these results creates a strong case for the MEI model of narrative comprehension and INFER's generation capability. The results of experiment 1 show that MEI's model of focus predicts measures of activation in readers. Participants were able to more quickly read sentences that were predicted as having higher relatedness by the MEI model. The results of experiment 2 show that MEI's model of inferences predicts changes in measures of activation of related concepts in readers. Participants in the fast reader group slowed significantly when deciding whether a word related to an inference was in the narrative (when, in fact, it was). The results of experiment 3 validated INFER's ability to generate Sequences of Discourse Content to prompt inferences. Participants were able to recognize a word in the narrative more quickly when it was related to a prompted inference. The sum of these results indicates that INFER is operating on a valid reader model, and is able to effectively use it's reader model to generate Sequences of Discourse Content with desired comprehension criteria.

Though these results provide a cogent argument for the theory behind the INFER system, more evaluation is always possible. In particular, it may be elucidating to observe how INFER's model of focus and inferencing interact with each other and other variables. Are inferences constructed more readily in high focus conditions? Does the length of the text change the comprehension effects? What are the specific effects of focus and inferencing on recall and summarization of narratives? Further experimentation may be used to answer these questions.

Chapter 5

Conclusion

The principle contributions of this work are an empirically evaluated formal model of narrative focus, an empirically evaluated formal model of narrative inferencing, and an empirically evaluated method of selecting narrative discourse content to satisfy comprehension criteria. This work is unique in that it 1) employs theory from cognitive models of narrative discourse comprehension to inform the creation of computational models of narrative comprehension and 2) employs these models for generation.

5.1 Summary

This dissertation defines an explicit computational model of a reader's comprehension process during reading, predicting aspects of narrative focus and inferencing. Focus is derived from activation values in an association network. Inferences about the story world are generated by a partial-order planner. This reader model is employed in a narrative discourse generation system to select content from an event log, creating discourses that satisfy comprehension criteria. The generation component defines a novel partial-order planning algorithm to satisfy the criteria. The results of this work have implications for both congitive scientists and builders of narrative generation systems.

I proposed a model of narrative comprehension, the MEI model. It is a formal, computational model of focus and inferencing in narrative comprehension consisting of the *Situation Model*, an activation network to compute focus, and the *Reader's Story*, an IPOCL plan to maintain the reader's understanding of the story. The situation model and Reader's Story are constructed incrementally as each element from the discourse is processed. The situation model maintains the connections between elements along the dimensions of space, time, causality, protagonists and objects, and intentions. It uses these connections to model how focus spreads from one element to the next during comprehension. The Reader's Story represents the reader's view of the story structure as the discourse unfolds. The Reader's Story is used as the starting point for generating inferences after each element of the discourse is processed.

Thus, the reader model makes three types of predictions, one for each type of comprehension criteria. These are: which inferences are prompted by the causal structure of the story, which inferences are prompted from the intentional structure of the story, and which elements are most salient in the reader's mind. These predictions are used to answer three types of questions during generation: which elements might be included to prompt for a causal inference, which elements might be included to prompt for an intentional inference, and which elements might be included to draw focus to a particular element.

The INFER system generates Sequences of Discourse Content that satisfy focus and inferencing comprehension criteria. INFER generates narrative discourses by searching through the space of possible Discourse Plans, solving one comprehension criterion at a time. The comprehension criteria are satisfied according to the predictions of the MEI reader model for inferencing necessitated by causal structure, inferencing motivated by intentional structure, and focus. The criteria for causal inferences specify a single step in an inference; the step must be part of a inference that is enabled and causally necessitated, and thus predicted by the reader model. Likewise, the criteria for intentional inferences specify a single step; the step must be part of an inference that is enabled and intentionally necessitated. The criteria for focus specifies a total activation for a node in the Situation Model. To achieve a focus criterion, the specified node must have a total activation at or above the specified amount over the course of the discourse.

A series of experiments were performed to evaluate the efficacy of the system. The first experiment measured the ability of the MEI model to predict comprehension effects due to narrative focus. The experimental results supported the focus model. The second experiment measured the ability of the MEI model to predict comprehension effects due to narrative inferencing. The experimental results show support for the inference model. The third experiment measured the ability for INFER to generate discourse to produce comprehension effects. The experimental results showed strong support for the effects obtained through generation. The sum of these results indicates that the MEI model is a valid reader model, and INFER able to effectively use this model to generate Sequences of Discourse Content with desired comprehension criteria.

This work has implications for both cognitive scientists and builders of narrative generation systems. For cognitive scientists, new, more precise models of narrative comprehension may be defined and tested, removing possible researcher bias and improving the independence of experimental measures. For implementers of systems, the ability to generate narrative discourses that are more coherent and cohesive may improve the effectiveness of narrative systems in entertainment, training, and education. The ability to predict aspects of comprehension in generated discourses gives the implementers new power for influencing narrative experiences.

5.2 Limitations

Although this work is unique in its approach, there are some limitations. First, although the computational complexity of the planning algorithm matches the complexity of the decision task presented by the problem, the complexity is still generally non-polynomial. This prevents INFER from generating long sequences of discourse content. Second, the specification of narrative content solely by the comprehension criteria is not natural to authors. Authors may think about the reader's comprehension, but few would specify focus and inference criteria as the sole means for determining content. This relates to the third limitation, the expressiveness of the reader model is limited to two main properties of comprehension, while cognitive models of narrative discourse comprehension have found evidence for many differing properties of comprehension. It is important to note, however, that the layered effect of many comprehension properties does not cancel the contribution of focus and inferencing in the average case. Fourth, the model has been implemented and tested by generating textual discourses, but many other narratives are available. Lastly, while the empirical evaluation of the presented model covered the three main contributions, the empirical evaluation of narrative generation systems is complex and difficult, and more experiments may provide further evidence for or against components of the system.

5.3 Future Work

There are several possible areas of future work. The computational complexity of INFER may be addressed by better heuristics or sacrificing some of the completeness of the generation algorithm. The reader model may be extended to account for more aspects of narrative comprehension: other forms of inference, the skill of the reader, situated reasoning models. The comprehension criteria may be modified to provide a more natural means of expressing authorial control over the reading experience, and the new requirements may be propagated back into the model. Expanding the target medium beyond text, perhaps to film, presents new and interesting challenges, and interactive narrative may call for comprehension models which are reactive to user input. Allowing for the reordering of content from the event log may require representations of discourse markers, to mark transitions between time segments. Finally, generating sequences of discourse content from a planning domain, rather than an event log, is an intriguing prospect. In such a system, the planner is responsible not only for choosing content at the discourse level, but also constructing a coherent story level to adhere to comprehension criteria.

5.4 Concluding Remarks

The means by which readers understand the story world and the events of the narrative are of primary interest when generating a narrative discourse. This dissertation describes a model of narrative focus, a model of narrative inference, and a method for selecting content from an event log to satisfy focus and inference comprehension criteria. By simulating these low level comprehension processes, higher level narrative attributes such as cohesion and coherence can be estimated. Generating narrative discourses to adhere to comprehension criteria increases their comprehensibility and effectiveness.

Bibliography

[1] Nicolas Szilas. A computational model of an intelligent narrator for interactive narratives. *Applied Artificial Intelligence*, 21(8):753–801, 2007.

[2] Michael Lebowitz. Story-telling as planning and learning. *Poetics*, 14, 1985.

[3] Michael Mateas and Andrew Stern. Faade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference, Game Design track*, March 2003. http://www.interactivestory.net.

[4] Mark O. Riedl and R. Michael Young. An intent-driven planner for multi-agent story generation. *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, New York*, 2004.

[5] Bradford Mott and James Lester. U-director: A decision-theoretic narrative planning architecture for storytelling environments. In *Proceedings of the Fifth International Conference on Autonomous Agents and Multiagent Systems*, pages 977–984, Hakodate, Japan, 2006.

[6] Yun Gyung Cheong. *A Computational Model of Narrative for Suspense.* PhD thesis, North Carolina State University, 2007.

[7] Peter Norvig. *A Unified Theory of Inference for Text Understanding.* PhD thesis, University of California, Berkeley, 1987.

[8] J.S. Penberthy and David Weld. Ucpop: a sound, complete, partial order planner for adl. In *Proceedings Third International Conference on Principles of Knowledge Representation and Reasoning (KR-92)*, pages 103–114, 1992.

[9] Arthur Graesser, Brent Olde, and Bianca Klettke. How does the mind construct and represent stories? In M. C. Green, J. J. Strange, and T. C. Brock, editors, *Narrative Impact : Social and Cognitive Foundations*. L. Erlbaum Associates, 2002.

[10] Rolf A. Zwaan and Gabriel A. Radvansky. Situation models in language comprehension and memory. *Psychological Bulletin*, 1998.

[11] Walter Kintsch. *Comprehension: A paradigm for cognition.* Cambridge University Press, Cambridge, MA, 1998.

[12] Walter Kintsch. The role of knowledge in discourse comprehension: A construction integration model. *Psychological Review*, 95:163–182, 1988.

[13] Mark C. Langston, Tom Trabasso, , and Joseph P. Magliano. A connectionist model of narrative comprehension. In *Understanding Language Understanding*. MIT Press, 1999.

[14] Tom Trabasso and Jennifer Wiley. Goal plans of action and inferences during comprehension of narratives. *Discourse Processes*, 39(2):129–164, May 2005.

[15] James R. Meehan. Tale-spin, an interactive program that writes stories. In *Fifth International Conference on Artificial Intelligence (IJCAI 77)*, 1977.

[16] David Pizzi, Marc Cavazza, and Jean-Luc Lugrin. Extending character-based storytelling with awareness and feelings. In *ACM Joint Conference on Autonomous Agents and Multi-Agent Systems*, Hawaii, May 2007. AAMAS.

[17] Marc Cavazza, Fred Charles, and Steven J. Mead. Planning characters' behaviour in interactive storytelling. *Journal of Visualization and Computer Animation*, 13:121–131, 2002.

[18] Randall W. Hill Jr., Jonathan Gratch, Stacy Marsella, Jeff Rickel, William Swartout, and David Traum. Virtual humans in the mission rehearsal exercise system. *KI special issue on Embodied Conversational Agents*, 2003.

[19] Ruth Aylett Sandy Louchart Joao Dias and Anna Paiva. Fearnot! an experiment in emergent narrative. In *Proceedings of the 5th International Conference on Intelligent Virtual Agents*, 2005.

[20] Nikitas Sgouros. Dynamic generation, management and resolution of interactive plots. *Artificial Intelligence*, 107:29–62, 1999.

[21] Scott R. Turner. *MINSTREL: A Computer Model of Creativity and Storytelling*. PhD thesis, University of California, Los Angeles, 1992.

[22] Yun-Gyung Cheong and R. Michael Young. Narrative generation for suspense: Modeling and evaluation. In *Proceedings of the International Conference on Interactive Digital Storytelling (ICIDS 08)*, Erfurt, Germany, November 2008.

[23] Byung-Chull Bae and R. Michael Young. A use of flashback and foreshadowing for surprise arousal in narrative using a plan-based approach. In *Proceedings of the International Conference on Interactive Digital Storytelling (ICIDS 08)*, Erfurt, Germany, November 2008.

[24] R. Michael Young, Mark O. Riedl, Mark Branly, R.J. Martin, and C.J. Saretto. An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development*, 1, 2004.

[25] Rolf A. Zwaan, Mark C. Langston, and Arthur Graesser. The construction of sitiation models in narrative comprehension: An event-indexing model. *Psychological Science*, 1995.

[26] Arthur Graesser, Murray Singer, and Tom Trabasso. Constructing inferences during narrative text comprehension. *Psychological Review*, 101(3):371–395, 1994.

[27] Gail McKoon and Roger Ratcliff. Inference during reading. *Psychological Review*, 99(3):440–466, 1992.

[28] Michael Mateas, Paul Vanouse, and Steffi Domike. Terminal time: An ideologically-biased history machine. *AISB Quarterly, Special Issue on Creativity in the Arts and Sciences*, 102:36–43, 1999.

[29] Michael Lebowitz. Creating characters in a story-telling universe. *Poetics*, 13:171–194, 1984.

[30] Michael Lebowitz. Planning stories. In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, Seattle, WA, 1987.

[31] Roger C. Schank and Robert P. Abelson. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures.* Erlbaum, Hillsdale, NJ, 1996.

[32] A. Bryan Loyall and Joseph Bates. Hap: A reactive, adaptive architecture for agents. Technical Report CMU-CS-91-147, School of Computer Science, Carnegie Mellon, Pittsburgh, PA, June 1991.

[33] Nicolas Szilas. Idtension: A narrative engine for interactive drama. *Proc. of the 1st Int. Conf. on Technologies for Interactive Digital Storytelling and Entertainment*, 2003.

[34] Mark O. Riedl. *Narrative Planning: Balancing Plot and Character.* PhD thesis, North Carolina State University, 2004.

[35] Mark O. Riedl and R. Michael Young. Story planning as exploratory creativity: Techniques for expanding the narrative search space. In *Proceedings of the 2005 IJCAI Workshop on Computational Creativity*, Edinburgh, 2005.

[36] Mark O. Riedl and Neha Sugandh. Story planning with vignettes: Toward overcoming the content production bottleneck. In *Proceedings of the 1st Joint International Conference on Interactive Digital Storytelling (Formerly TIDSE and ICVS)*, Erfurt, Germany, 2008.

[37] R. Michael Young and Martha E. Pollack. Decomposition and causality in partial-order planning. In *Proceedings of the Second International Conference on Artificial Intelligence and Planning Systems, Chicago, IL*, 1994.

[38] Gerald DeJong. Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3(3):251–273, 1979.

[39] Jerry R. Hobbs, Mark Stickel, Paul Martin, and Douglas Edwards. Interpretation as abduction. In *Proceedings of the 26th annual meeting on Association for Computational Linguistics*, pages 95–103, Morristown, NJ, USA, 1988. Association for Computational Linguistics.

[40] SC Shapiro and WJ Rapaport. An introduction to a computational reader of narratives. In JF Duchan, GA Bruder, and Hewitt LE, editors, *Deixis in Narrative: A Cognitive Science Perspective*, pages 79–105. Lawrence Erlbaum, Hillsdale, NJ, 1995.

[41] Stefan L. Frank, Mathieu Koppenc, Leo G. M. Noordmana, and Wietske Vonk. Modeling knowledge-based inferences in story comprehension. *Cognitive Science*, 27:875–910, 2003.

[42] Ashwin Ram and Kenneth Moorman, editors. *Understanding Language Understanding: Computional Models of Reading.* MIT Press, 1999.

[43] Erik T. Mueller. Story understanding. In *Encyclopedia of Cognitive Science*, volume 4, pages 238–46. Nature Publishing Group, London, 2002.

[44] Roger Schank. Sam: A story understander. Technical Report 46, Yale University, 1975.

[45] Rich Cullingford. Sam: a program that uses world knowledge to understand. *ACM SIGART Bulletin*, 61:53–54, 1977.

[46] Robert Wilensky. Using plans to understand natural language. In *ACM 76: Proceedings of the annual conference*, pages 46–50, New York, NY, USA, 1976. ACM Press.

[47] Peter Norvig. Inference in text understanding. In *Proceedings of the National Conference on Artificial Intelligence*, pages 561–565, 1987.

[48] Erik T. Mueller. Understanding goal-based stories through model finding and planning. In Brian Magerko and Mark Riedl, editors, *Intelligent Narrative Technologies: Papers from the AAAI Fall Symposium*, pages 95–101, Menlo Park, CA, 2007. AAAI Press.

[49] Erik T. Mueller. Story understanding through multi-representation model construction. In *Text Meaning: Proceedings of the HLT-NAACL 2003 Workshop*, 2003.

[50] Walter Kintsch Teun Adrianus van Dijk. *Strategies of Discourse Comprehension.* Academic Press, 1983.

[51] Michael Rinck. Temporal and spatial distance in situation models. *Memory and cognition*, 28:1310–1320, 2000.

[52] Joseph P. Magliano, Jason Miller, and Rolf A. Zwaan. Indexing space and time in film understanding. *Applied Cognitive Psychology*, 15(5):533–545, 2001.

[53] Joseph P. Magliano, Holly A. Taylor, and Hyun-Jeong Joyce Kim. When goals collide: Monitoring the goals of multiple characters. *Memory and Cognition*, 33(8):1357–1367, 2005.

[54] David J. Therriault and Gary E. Raney. Further exploration of time in situation models. evidence for the representation of duration information. *Discourse Processes*, 43(2):173–200, 2007.

[55] Lei Mo, Hongmin Chen, Ying Li, Zhe Chen, and Xianyou He. Effects of event-related centrality on concept accessibility. *Discourse Processes*, 43(3), 2007.

[56] Tracy Linderholm. Predictive inference generation as a function of working memory capacity and causal text constraints. *Discourse Processes*, 34(3):259–280, Nov 2002.

[57] David Allbritton. Strategic production of predictive inferences during comprehension. *Discourse Processes*, 38(3):309–322, Nov 2004.

[58] Connie Shears, Vanessa Miller, Megan Ball, Amanda Hawkins, Janna Griggs, and Andria Varner. Cognitive demand differences in causal inferences: Characters' plans are more difficult to comprehend than physical causation. *Discourse Processes*, 43(3):255–278, May 2007.

[59] Mark A. Casteel. Contextual support and predictive inferences: What do readers generate and keep available for use. *Discourse Processes*, 44(1):51–72, Jul 2007.

[60] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersy, second edition, 2003.

[61] Tom Bylander. The computational complexity of propositional strips planning. *Artificial Intelligence*, 69:165–204, 1994.

[62] Richard Fikes and Nils J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[63] Edwin P.D. Pednault. Formulating multiagent, dynamic-world problems in the classical planning framework. In *Reasoning about Actions and Plans: Proceedings of the 1986 Workshop*, pages 47–82. Morgan Kaufmann, Timberline, Oregon, 1986.

[64] Richard Gerrig. *Experiencing narrative worlds on the psychological activities of reading*. Yale University Press, New Haven, CT, 1993.

[65] Bradford Mott, Sunyoung Lee, and James Lester. Probablistic goal recognition in interactive narrative environments. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 187–192, Boston, Massachusetts, 2006.

[66] Sunyoung Lee, Scott McQuiggan, and James Lester. Inducing user affect recognition models for task-oriented environments. In *Proceedings of the Eleventh International Conference on User Modeling*, pages 380–384, Corfu, Greece, 2007.

[67] Scott McQuiggan and James Lester. Modeling and evaluating empathy in embodied companion agents. *International Journal of Human-Computer Studies*, 65(4):348–360, 2007.

[68] Mark O. Riedl and R. Michael Young. An intent-driven planner for multi-agent story generation. *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, New York*, 2004.

[69] William J. Stewart. *Introduction to the numerical solution of Markov chains.* Princeton University Press, 1994.

[70] Arthur Graesser and Leslie Clark. *Structures and procedures of implicit knowledge.* Ablex, Norwood, NJ, 1985.

[71] J. L. Myers, M. Shinjo, and S. A. Duffy. Degrees of causal relatedness and memory. *Journal of Memory and Language*, 26:453–465, 1987.

[72] Arthur Graesser and Eugenie L. Bertus. The construction of causal inferences while reading expository texts on science and technology. *Scientific Studies of Reading*, 2:247–269, 1998.

[73] Soyoung Suh and Tom Trabasso. Inferences during reading: Converging evidence from discourse analysis, talk-aloud protocols, and recognition priming. *Journal of Memory and Language*, 32:279–300, 1993.

Appendices

Appendix A, Experimental Materials

Experiment 1

The experment's software was implemented as a Java Applet executed in a browser window. Section 5.4 presents screenshots of this applet with the instructions and materials for Experiment 3. The instructions and materials for Experiment 1 are below.

Instructions

Beginning instructions:

```
In this study you will be asked to read twelve short stories divided
into three sets.  The stories will be shown to you one sentence at a
time.  Once you are finished reading each sentence, press the left mouse
button to continue to the next line.  You cannot go back or repeat sentences
you've already read.  Please read these stories for comprehension.

After you have finished each story, you will be asked two questions on
whether words or phrases appeared in the story you just read.
We will start with a short example.
```

Before story instructions:

```
Beginning of Story.  Click the left mouse button to begin reading.
```

Before Word Recognition instructions:

```
Word recognition questions.  Click the left mouse button to continue.
```

Materials

```
There was a boy named Billy.
Billy liked flowers.
Billy's Mom gave Billy some money.
Billy's Mom told Billy to buy the cheese at the grocery.
Billy started walking to the grocery.
Billy saw the cheese at the grocery.
Billy bought the cheese at the grocery.
Billy started walking to home.
Billy gave Billy's Mom the cheese.
```

Figure A.1: Story 1, High Relatedness

```
There was a boy named Billy.
Billy liked flowers.
Billy's Mom gave Billy some money.
Billy's Mom told Billy to buy the cheese at the grocery.
Billy started walking to the grocery.
Billy saw flowers at the florist.
Billy bought the cheese at the grocery.
Billy started walking to home.
Billy gave Billy's Mom the cheese.
```

Figure A.2: Story 1, Low Relatedness

```
There was a girl named Betty.
Betty needed some detergent.
Betty was friends with Sue.
Sue had a birthday coming soon.
Sue liked perfume.
Betty wanted Sue to have perfume.
The corner store sold perfume.
Betty went from Betty's home to the corner store.
Betty bought perfume at the corner store.
Betty attended the birthday party for Sue.
Betty gave Sue perfume.
Sue had a good time.
```

Figure A.3: Story 2, High Relatedness

```
There was a girl named Betty.
Betty needed some detergent.
Betty was friends with Sue.
Sue had a birthday coming soon.
Sue liked perfume.
Betty wanted Sue to have perfume.
The corner store sold perfume.
Betty went from Betty's home to the corner store.
Betty bought detergent at the corner store.
Betty attended the birthday party for Sue.
Betty gave Sue perfume.
Sue had a good time.
```

Figure A.4: Story 2, Low Relatedness

```
There was a man named Stan.
Stan worked at a paper company.
There was a woman named Lisa.
Lisa worked with Stan.
Lisa went from the office to the salon.
Stan wanted to be promoted.
The boss asked Stan to present the sales earnings.
Stan wanted to do a good job on the presentation.
Stan went from Stan's home to the office.
Stan read a book at the office about presentations.
Stan did a good job presenting the the sales earnings.
Stan recieved a promotion.
```

Figure A.5: Story 3, High Relatedness

```
There was a man named Stan.
Stan worked at a paper company.
There was a woman named Lisa.
Lisa worked with Stan.
Lisa went from the office to the salon.
Stan wanted to be promoted.
The boss asked Stan to present the sales earnings.
Stan wanted to do a good job on the presentation.
Stan went from Stan's home to the office.
Lisa read a book at the salon about presidents.
Stan did a good job presenting the the sales earnings.
Stan received a promotion.
```

Figure A.6: Story 3, Low Relatedness

```
There was a knight named Jacob.
There was a princess named Elaine.
There was an evil warlord named Dylan.
Jacob loved Elaine.
Dylan was dirty.
Dylan drew a bath at Dylan's chambers.
Dylan kidnaped Elaine.
Dylan locked Elaine in the jail atop the tower.
Jacob rode from Jacob's court to the tower.
Jacob climbed the tower.
Jacob stepped into the top of the tower.
Jacob kicked down the door holding Elaine.
Jacob rescued Elaine.
```

Figure A.7: Story 4, High Relatedness

```
There was a knight named Jacob.
There was a princess named Elaine.
There was an evil warlord named Dylan.
Jacob loved Elaine.
Dylan was dirty.
Dylan drew a bath at Dylan's chambers.
Dylan kidnaped Elaine.
Dylan locked Elaine in the jail atop the tower.
Jacob rode from Jacob's court to the tower.
Jacob climbed the tower.
Dylan stepped into the back of the bath.
Jacob kicked down the door holding Elaine.
Jacob rescued Elaine.
```

Figure A.8: Story 4, Low Relatedness

Experiment 2

The experment's software was implemented as a Java Applet executed in a browser window. Section 5.4 presents screenshots of this applet with the Experiment 3 instructions and materials. The instructions and materials for Experiment 2 are below.

Instructions

Beginning instructions:

```
In this study you will be asked to read twelve short stories divided
into three sets.  The stories will be shown to you one sentence at a
time.  Once you are finished reading each sentence, press the spacebar
to continue to the next line.  You cannot go back or repeat sentences
you've already read.  Please read these stories for comprehension.

After you have finished each story, you will be asked two questions on
whether words or phrases appeared in the story you just read.  We will
start with a short example.
```

Before story instructions:

```
Ready for Story.  Press the spacebar to begin reading.
```

Before Word Recognition instructions:

```
Ready for Word Recognition

Place your hands on the keyboard such that

1) the index finger on your left hand is on the 'z' key
2) the index finger on your right hand is on the '/?' key
```

Materials

Causally Necessitated

```
Jimmy played guitar in a band.
Ben, Sandra, and Nathan played in the band with Jimmy.
Jimmy's guitar was stolen.
Jimmy wanted a new guitar.
Ben offered Jimmy an old guitar.
Jimmy saved all of his money in the bank.
Jimmy went into the guitar shop.
Jimmy came out of the guitar shop.
Jimmy played the guitar.
Jimmy went to the studio.
Sandra wanted to drive Nathan and Ben from Nathan's house to the studio.
Sandra drove the van to Nathan's house.
Nathan and Ben got in the van.
The van's tire was flat.
Sandra got the spare tire from the trunk.
Nathan and Ben propped up the van.
Sandra changed the tire.
Sandra, Nathan, and Ben drove to the studio.
Jimmy, Sandra, Nathan, and Ben jammed out.
```

Figure A.9: Causally Necessitated Story 1, Prompted

```
Jimmy played guitar in a band.
Ben, Sandra, and Nathan played in the band with Jimmy.
Jimmy's guitar was stolen.
Jimmy wanted a new guitar.
Ben offered Jimmy an old guitar.
Jimmy lost all of his money in the stock market.
Jimmy went into the guitar shop.
Jimmy came out of the guitar shop.
Jimmy got into his car.
Jimmy went to the studio.
Sandra wanted to drive Nathan and Ben from Nathan's house to the studio.
Sandra drove the van to Nathan's house.
Nathan and Ben got in the van.
The van's tire was flat.
Sandra got the spare tire from the trunk.
Nathan and Ben propped up the van.
Sandra changed the tire.
Sandra, Nathan, and Ben drove to the studio.
Jimmy, Sandra, Nathan, and Ben jammed out.
```

Figure A.10: Causally Necessitated Story 1, Un-prompted

```
bought
drove
```

Figure A.11: Word Recognition words for Causally Necessitated Story 1, Prompted and Un-prompted

```
Dave was eating some bacon in the kitchen.
Bogie was Dave's dog.
Vicky was married to Dave.
Alice was friends with Vicky.
There was an expensive vase in the living room.
Vicky was reading the newspaper in the kitchen.
Vicky read a newspaper article about Alice.
Bogie wanted some bacon.
Bogie went to the living room.
There was a loud sound in the living room.
Bogie went to the kitchen.
Dave went to the living room.
Dave saw a shattered vase.
Dave went to the kitchen.
The bacon was gone.
Vicky wanted to tell Alice about the newspaper article.
Vicky called Alice.
Alice was excited to hear about the newspaper article.
Dave wanted to buy more bacon.
Dave went to the grocery store.
Vicky went to a bar to celebrate with Alice.
```

Figure A.12: Causally Necessitated Story 2, Prompted

```
Dave was eating some bacon in the kitchen.
Bogie was Dave’s dog.
Vicky was married to Dave.
Alice was friends with Vicky.
There was an expensive vase in the living room.
Vicky was reading the newspaper in the kitchen.
Vicky read a newspaper article about Alice.
Bogie wanted some bacon.
Bogie went to the living room.
There was a loud sound in the living room.
Bogie went to the kitchen.
Dave went to the living room.
Dave saw a music video.
Dave went to the kitchen.
The bacon was gone.
Vicky wanted to tell Alice about the newspaper article.
Vicky called Alice.
Alice was excited to hear about the newspaper article.
Dave wanted to buy more bacon.
Dave went to the grocery store.
Vicky went to a bar to celebrate with Alice.
```

Figure A.13: Causally Necessitated Story 2, Un-prompted

```
knocked over
whined
```

Figure A.14: Word Recognition words for Causally Necessitated Story 2, Prompted and Un-prompted

```
Jessica lived in the blue house.
The blue house was next to the river.
The blue house had a clean, beautiful lawn.
The blue house had clean, beautiful blue wallpaper.
The blue house had clean, beautiful white carpet.
Jessica was friends with Emma.
Jessica went to visit Emma.
It rained heavily at the blue house.
The sun came out at the blue house.
Jessica went to the blue house.
The lawn was covered in mud.
The blue wallpaper was soggy.
The white carpet was brown and wet.
Jessica was sad.
Jessica wanted to play soccer that night.
Jessica's soccer uniform was dirty.
Jessica took Jessica's soccer uniform to the laundromat.
Jessica saw Emma at the laundromat.
Emma made Jessica feel better.
Jessica played soccer that night.
Jessica scored a goal.
```

Figure A.15: Causally Necessitated Story 3, Prompted

```
Jessica lived in the blue house.
The blue house was next to the river.
The blue house had a clean, beautiful lawn.
The blue house had clean, beautiful blue wallpaper.
The blue house had clean, beautiful white carpet.
Jessica was friends with Emma.
Jessica went to visit Emma.
It rained heavily at the blue house.
The sun came out at the blue house.
Jessica went to the blue house.
The lawn was slightly damp.
The blue wallpaper was sparkling.
The white carpet was fluffy.
Jessica was sad.
Jessica wanted to play soccer that night.
Jessica's soccer uniform was dirty.
Jessica took Jessica's soccer uniform to the laundromat.
Jessica saw Emma at the laundromat.
Emma made Jessica feel better.
Jessica played soccer that night.
Jessica scored a goal.
```

Figure A.16: Causally Necessitated Story 3, Un-prompted

```
visit
flooded
```

Figure A.17: Word Recognition words for Causally Necessitated Story 3, Prompted and Un-prompted

```
Bill, Isabel, and Thomas were playing golf.
The large hill was next to the hole.
The lake was next to the hole.
The lake was smooth and quiet.
Bill wanted to knock the ball in the hole.
Bill hit Bill's ball with a driver.
Bill's ball went over the large hill.
Bill went to the top of the large hill.
Bill saw ripples on the lake.
Bill did not see Bill's ball.
Isabel hit Isabel's ball with a driver.
Isabel's ball went over the large hill.
Bill saw Isabel's ball near the hole.
Thomas hit Thomas's ball with a driver.
Thomas's ball went over the large hill.
Bill saw Thomas's ball go into the hole.
Bill, Isabel, and Thomas went to the club house.
Thomas bought a round of drinks.
Bill did not like Thomas.
Bill drank some soda.
```

Figure A.18: Causally Necessitated Story 4, Prompted

```
Bill, Isabel, and Thomas were playing golf.
The large hill was next to the hole.
The lake was next to the hole.
The lake was smooth and quiet.
Bill wanted to knock the ball in the hole.
Bill hit Bill's ball with a driver.
Bill's ball went over the large hill.
Bill went to the top of the large hill.
Bill saw ripples on the lake.
Bill saw Bill's ball in the woods.
Isabel hit Isabel's ball with a driver.
Isabel's ball went over the large hill.
Bill saw Isabel's ball near the hole.
Thomas hit Thomas's ball with a driver.
Thomas's ball went over the large hill.
Bill saw Thomas's ball go into the hole.
Bill, Isabel, and Thomas went to the club house.
Thomas bought a round of drinks.
Bill did not like Thomas.
Bill drank some soda.
```

Figure A.19: Causally Necessitated Story 4, Un-prompted

```
splashed
wind
```

Figure A.20: Word Recognition words for Causally Necessitated Story 4, Prompted and Un-prompted

```
There was a prince named Olive.
There was an evil dragon named Draco.
There was a roaming hero named Ava.
Draco wanted to cook and eat Olive.
Draco did not like raw food.
Draco kidnapped Olive.
Draco took Olive to Draco's cave.
There was a cauldron at Draco's cave.
Draco lit a fire under the cauldron.
The cauldron started to boil.
Draco lifted Olive above the cauldron.
Ava ran into Draco's cave.
Ava drew Ava's sword.
Ava slashed at Draco with Ava's sword.
Draco blew fire at Ava.
The drapes in the cave caught fire.
Ava began to cough at the smoke.
Ava grabbed Olive.
Ava and Olive ran out of the cave.
Ava and Olive rode into the sunset.
```

Figure A.21: Intentionally Necessitated Story 1, Prompted

Intentionally Necessitated

```
There was a prince named Olive.
There was an evil dragon named Draco.
There was a roaming hero named Ava.
Draco wanted to cook and eat Olive.
Draco did not like raw food.
Draco kidnapped Olive.
Draco took Olive to Draco’s cave.
There was a cauldron at Draco’s cave.
The dragon swept the dust under the cauldron.
The dust started to settle.
The dragon straightened his drapes.
Ava ran into Draco’s cave.
Ava drew Ava’s sword.
Ava slashed at Draco with Ava’s sword.
Draco blew fire at Ava.
The drapes in the cave caught fire.
Ava began to cough at the smoke.
Ava grabbed Olive.
Ava and Olive ran out of the cave.
Ava and Olive rode into the sunset.
```

Figure A.22: Intentionally Necessitated Story 1, Un-prompted

```
cook
blew
```

Figure A.23: Word Recognition words for Intentionally Necessitated Story 1, Prompted and Un-prompted

```
There was a man named Ed.
William and Madison were Ed's friends.
William gave Ed an alligator for his birthday.
The alligator was Ed's favorite present.
Ed took the alligator to Ed's apartment.
Ed's landlord knocked on Ed's door.
Ed wanted to hide the alligator.
Ed put the alligator under the bed.
The alligator's tail was sticking out.
Ed went to the laundry room.
Ed found a blanket and a pair of scissors.
Ed was out of tape.
Madison called Ed on the phone.
Ed invited Madison to a restaurant for dinner.
Ed and Madison went to the restaurant.
Ed and Madison ate dinner.
Ed went to the hardware store.
Ed bought some tape.
Ed went to Ed's apartment.
```

Figure A.24: Intentionally Necessitated Story 2, Prompted

```
There was a man named Ed.
William and Madison were Ed's friends.
William gave Ed an alligator for his birthday.
The alligator was Ed's favorite present.
Ed took the alligator to Ed's apartment.
Ed's landlord knocked on Ed's door.
Ed wanted to hide the alligator.
Ed put the alligator under the bed.
Ed wanted to make some tea.
Ed went to the kitchen.
Ed got some water and a tea kettle.
Ed was out of tea.
Madison called Ed on the phone.
Ed invited Madison to a restaurant for dinner.
Ed and Madison went to the restaurant.
Ed and Madison ate dinner.
Ed went to the grocery store.
Ed bought some tea.
Ed went to Ed's apartment.
```

Figure A.25: Intentionally Necessitated Story 2, Un-prompted

```
hide
gave
```

Figure A.26: Word Recognition words for Intentionally Necessitated Story 2, Prompted and Un-prompted

```
Julia and Sophia had dessert.
Julia got a toothache.
Julia wanted to get the tooth pulled.
Julia went to the dentist's office.
The receptionist asked Julia to make an appointment.
Julia made an appointment.
Julia and Sophia went to the mall.
Julia and Sophia went to the clothing store.
Julia tried on new suits.
Sophia tried on new party dresses.
Julia and Sophia went to the food court.
Julia drank a diet soda.
Sophia ate a chili dog with extra mustard.
Sophia got mustard on Sophia's clothes.
Sophia bought a new outfit.
Julia and Sophia drove away from the mall.
Julia drove to the dentist's office.
```

Figure A.27: Intentionally Necessitated Story 3, Prompted

```
Julia and Sophia had dessert.
Julia got a toothache.
Julia wanted to get the tooth pulled.
Julia went to the dentist's office.
The receptionist asked Julia to take a seat.
The dentist saw Julia.
Julia and Sophia went to the mall.
Julia and Sophia went to the clothing store.
Julia tried on new suits.
Sophia tried on new party dresses.
Julia and Sophia went to the food court.
Julia drank a diet soda.
Sophia ate a chili dog with extra mustard.
Sophia got mustard on Sophia's clothes.
Sophia bought a new outfit.
Julia and Sophia drove away from the mall.
Julia drove past the dentist's office.
```

Figure A.28: Intentionally Necessitated Story 3, Un-prompted

```
pulled
swam
```

Figure A.29: Word Recognition words for Intentionally Necessitated Story 3, Prompted and Un-prompted

```
The dictator arrived at the podium.
The crowd cheered.
The assassin wanted to kill the dictator.
The assassin reached for the rifle.
The assassin lifted the rifle to his shoulder.
The assassin peered through the scope.
The scope fell off the rifle.
The assassin lay prone to draw a sight without a scope.
The sun blinded the assassin.
```

Figure A.30: Intentionally Necessitated Story 4, Prompted

```
The dictator arrived at the podium.
The crowd cheered.
The assassin wanted to kill the dictator.
The assassin reached for the rifle.
The assassin lifted the rifle to his shoulder.
The assassin peered through the scope.
The assassin shot the dictator.
The assassin started to run towards the west.
The sun blinded the assassin.
```

Figure A.31: Intentionally Necessitated Story 4, Un-prompted

```
kill
rifle
```

Figure A.32: Word Recognition words for Intentionally Necessitated Story 4, Prompted and Un-prompted

Experiment 3

The experment's software was implemented as a Java Applet executed in a browser window. Screenshots of this applet with the Experiment 3 instructions and materials are below.

Instructions and ScreenShots

Beginning instructions:

```
In this study you will be asked to read twelve short stories divided
into three sets.  The stories will be shown to you one sentence at a
time.  Once you are finished reading each sentence, press enter
to continue to the next line.  You cannot go back or repeat sentences
you've already read.  Please read these stories for comprehension.

After you have finished each story, you will be asked one comprehension
question and 5 questions on whether words or phrases appeared in the
story you just read.  After the story section, there will be an additional
series of questions.  Please answer these questions to the best of your
ability.

You will be timed while reading or answering questions, so please do not
take breaks except where indicated.

We will start with a short example.
```

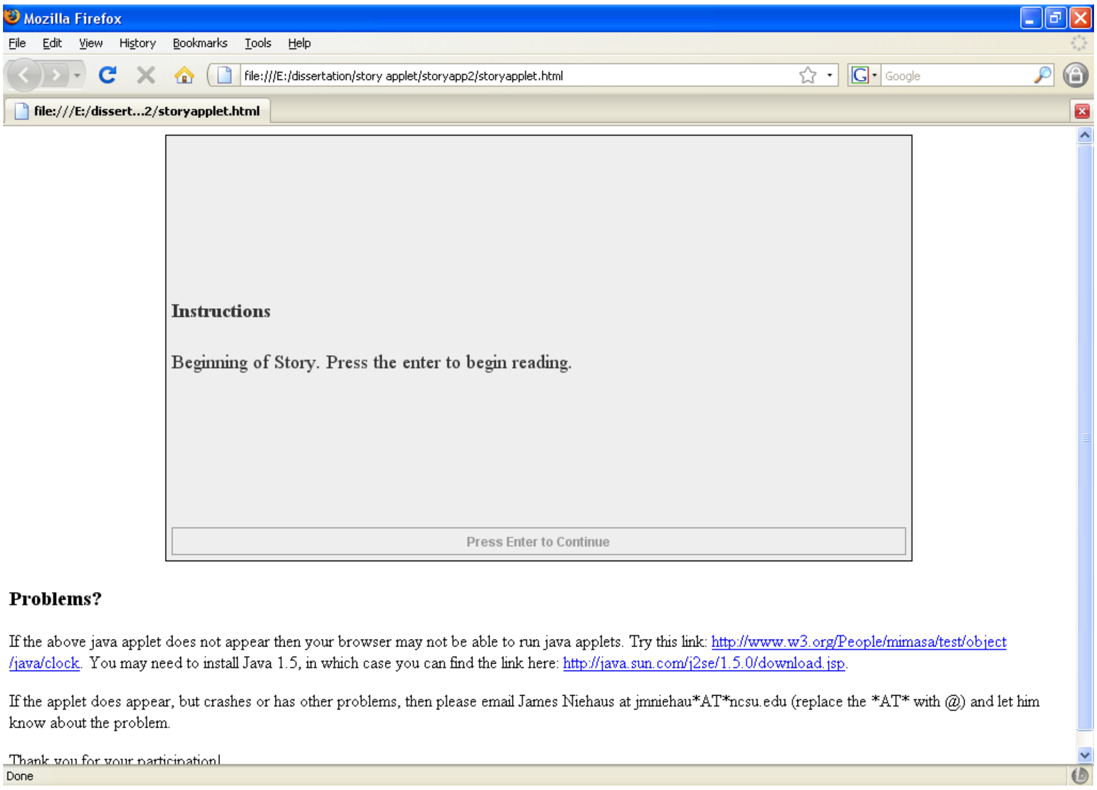

Figure A.33: Experiment Java Applet running in a browser window.

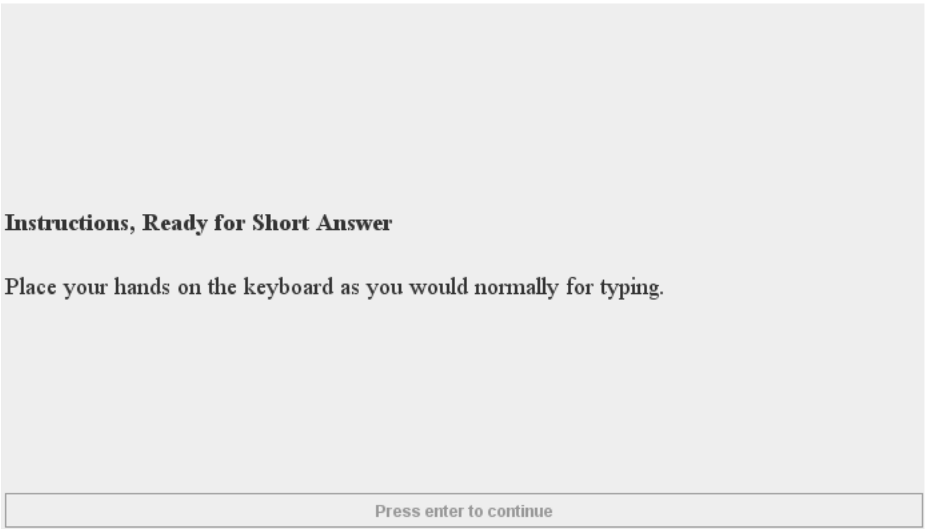

Figure A.34: Instructions for Short Answer questions.

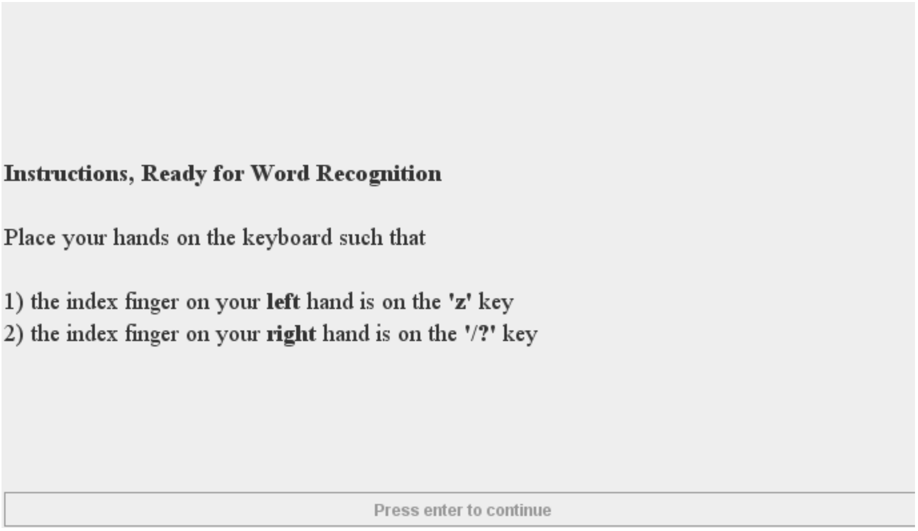

Figure A.35: Instructions for Word Recognition questions.

Figure A.36: Instructions for beginning a story.

Figure A.37: Reading a story.

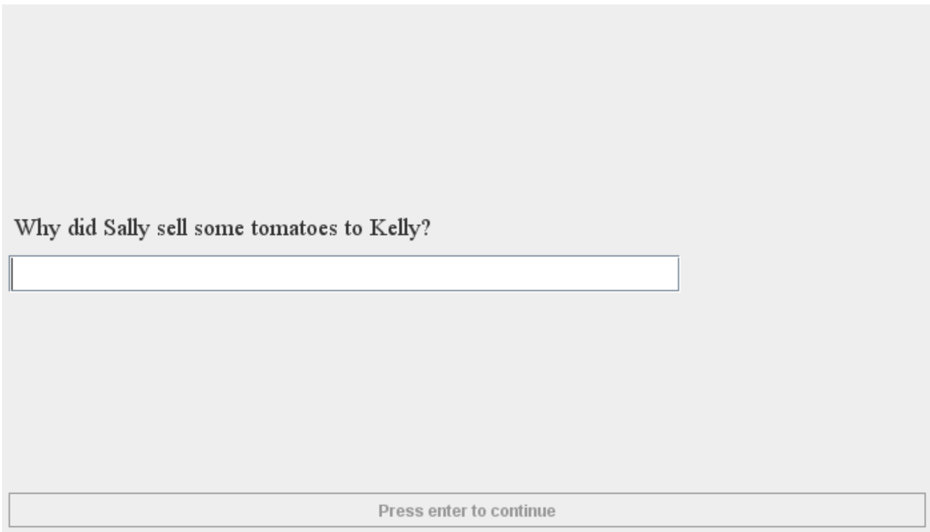

Figure A.38: Answering a Short Answer question.

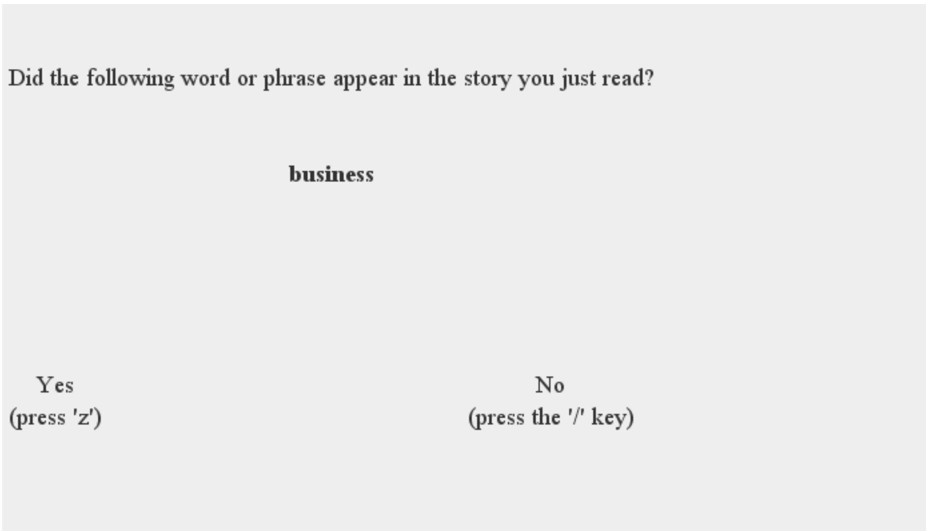

Figure A.39: Answering a Word Recognition question.

Materials

Source Event Log (IPOCL Plan) Planning Domain Definition:

```
;; pick something up
(define (action pick-up)
    :parameters (?person ?thing ?place)
    :actors (?person)
    :constraints ((person ?person) (thing ?thing) (place ?place))
    :precondition ((at ?person ?place) (at ?thing ?place))
    :primitive t
    :text-template ("?Person picked up ?thing at ?place."
    (subj "?person"))
    :effect ((:not (at ?thing ?place))
     (has ?person ?thing)))

;; pick up and holster a gun
(define (action hoslter-gun)
    :parameters (?person ?gun ?place)
    :actors (?person)
    :constraints ((person ?person) (gun ?gun) (place ?place))
    :precondition ((at ?person ?place) (at ?gun ?place))
    :primitive t
    :text-template ("?Person took ?thing from ?place and
                     put it into the holster on (?person hip)."
    (subj "?person"))
    :effect ((:not (at ?gun ?place))
     (has ?person ?gun)))

;; withdraw some money from the bank
(define (action withdraw-money)
    :parameters (?person ?bank ?money)
    :actors (?person)
    :constraints ((person ?person) (bank ?bank) (money ?money))
    :precondition ((at ?person ?bank) (has ?bank ?money))
    :primitive t
    :text-template ("?Person withdrew some money at ?bank
                     and put it in her purse." (subj "?person"))
    :effect ((:not (has ?bank ?money))
     (has ?person ?money)))

;; open a store for business
(define (action open)
    :parameters (?person ?store)
```

```
    :not-needs-intention t
    :constraints ((person ?person) (store ?store))
    :precondition ((at ?person ?store))
    :primitive t
    :text-template ("?Person opened ?store for business."
    (subj "?person"))
    :effect ((open-store ?store)))

;; sell some small goods
(define (action sell)
    :parameters (?person ?buyer ?thing ?money ?place)
    :actors (?person)
    :constraints ((person ?person) (person ?buyer)
  (small-goods ?thing) (money ?money)
  (place ?place))
    :precondition ((at ?person ?place) (has ?person ?thing)
   (has ?buyer ?money))
    :primitive t
    :text-template ("?Person sold ?thing to ?buyer for a little extra money."
    (subj "?person"))
    :effect ((:not (has ?person ?thing))
     (has ?person ?money)
     (:not (has ?buyer ?money))
     (has ?buyer ?thing)))

;; buy a dress (or other good) from a store
(define (action buy-dress)
    :parameters (?person ?thing ?store ?money)
    :actors (?person)
    :constraints ((person ?person) (store ?store) (money ?money)
  (place ?store) (thing ?thing) (valuable ?thing))
    :precondition ((open ?store)
   (forsale ?thing ?store)
   (at ?person ?store) (has ?person ?money))
    :primitive t
    :text-template ("?Person bought ?thing from ?store with ?money."
    (subj "?person"))
    :effect ((has ?person ?thing)
     (:not (forsale ?thing ?store))
     (:not (has ?person ?money))))


;; fail to buy a dress (or other good) from a store because of no money
(define (action fail-buy-dress)
```

```
    :parameters (?person ?thing ?store ?money)
    :actors (?person)
    :constraints ((person ?person) (store ?store) (money ?money)
  (place ?store) (thing ?thing) (valuable ?thing))
    :precondition ((open ?store)
   (forsale ?thing ?store)
   (at ?person ?store) (:not (has ?person ?money)))
    :primitive t
    :text-template ("?Person tried to buy ?thing from ?store,
                     but ?person didn't have ?money." (subj "?person"))
    :effect nil)

;; kick someone out of the way
(define (action kick-out-of-way)
    :parameters (?person ?roadblock ?alley ?place)
    :actors (?person)
    :constraints ((person ?person) (evil ?person)
  (person ?roadblock) (alley ?alley) (place ?place))
    :precondition ((at ?person ?place) (at ?roadblock ?place)
   (blocking ?roadblock ?alley))
    :primitive t
    :text-template ("?Person kicked at ?roadblock at ?place."
    (subj "?person"))
    :effect ((:not (blocking ?roadblock ?alley))))

;; hatch a plan to rob the bank
(define (action hatch-plan)
    :parameters (?person ?gun ?horse ?bank ?mother-lode)
    :not-needs-intention t
    :constraints ((person ?person) (evil ?person) (gun ?gun)
  (horse ?horse) (mother-lode ?mother-lode))
    :precondition ((has ?bank ?mother-lode))
    :primitive t
    :text-template ("?Person hatched an evil plan to rob the bank.
                     ?Person needed a gun and a horse." (subj "?person"))
    :effect (    (:not (has ?person ?mother-lode))
     (intends ?person (has ?person ?gun))
     (intends ?person (has ?person ?horse))
     (intends ?person (has ?person ?mother-lode))
     (intends ?person (free-with-money ?person))))

;; hide in an alley
(define (action hide-in-dark-alley)
    :parameters (?person ?alley)
```

```
    :actors (?person)
    :constraints ((person ?person) (evil ?person) (alley ?alley))
    :precondition ((at ?person ?alley))
    :primitive t
    :text-template ("?Person hid in ?alley." (subj "?person"))
    :effect ((hidden ?person)))

;; pickpocket
(define (action pickpocket)
    :parameters (?person ?mark ?money ?place ?alley)
    :actors (?person)
    :constraints ((person ?person) (evil ?person) (place ?place)
  (place ?alley) (alley ?alley) (alley-of ?alley ?place)
  (person ?mark) (money ?money))
    :precondition ((at ?person ?alley) (at ?mark ?place)
   (hidden ?person) (has ?mark ?money))
    :primitive t
    :text-template ("?Person picked the pocket of ?mark."
    (subj "?person"))
    :effect ((has ?person ?money) (:not (has ?mark ?money))
     (:not (hidden ?person))))

;; move
(define (action move-once)
    :parameters (?person ?from ?to)
    :actors (?person)
    :constraints ((mobile-person ?person) (place ?from) (place ?to)
  (connected ?from ?to) (connected ?to ?from))
    :precondition ((at ?person ?from))
    :primitive t
    :text-template ("?Person went from ?from to ?to." (subj "?person"))
    :effect ((at ?person ?to) (:not (at ?person ?from))))

;; buy drinks for (and get drunk)
(define (action buy-drinks-for)
    :parameters (?person ?drinker ?money ?place)
    :actors (?person)
    :constraints ((person ?person) (place ?place)
  (bar ?place) (person ?drinker) (money ?money))
    :precondition ((at ?person ?place) (at ?drinker ?place)
   (has ?person ?money))
    :primitive t
    :text-template ("?Person bought drinks for ?drinker, who got very drunk."
    (subj "?person"))
```

```
    :effect ((friendly ?drinker ?person) (drunk ?drinker)))

;; cheat at a poker game (put up some money)
(define (action cheat-at-poker)
    :parameters (?person ?poker ?money ?winnings ?place)
    :actors (?person)
    :constraints ((person ?person) (evil ?person) (poker-game ?poker)
  (place ?place) (money ?winnings) (money ?money))
    :precondition ((at ?person ?place) (at ?poker ?place)
   (has ?person ?money) (bet-at ?winnings ?poker))
    :primitive t
    :text-template ("?Person cheated at ?poker." (subj "?person"))
    :effect ((has ?person ?winnings)))

;; leave with
(define (action escort-drunk-friend)
    :parameters (?person ?friend ?from ?to)
    :actors (?person)
    :constraints ((person ?person) (place ?from) (place ?to)
  (person ?friend) (connected ?from ?to))
    :precondition ((at ?person ?from) (at ?friend ?from)
   (drunk ?friend) (friendly ?friend ?person))
    :primitive t
    :text-template ("?Person escorted ?friend from ?from to ?to."
    (subj "?person"))
    :effect ((:not (at ?person ?from)) (at ?person ?to)
     (:not (at ?friend ?from)) (at ?friend ?to)))

;; lay to rest in alley
(define (action lay-to-rest-in-alley)
    :parameters (?person ?friend ?place ?bank)
    :actors (?person)
    :constraints ((person ?person) (place ?place)
  (alley ?place) (person ?friend) (guard ?friend)
  (bank ?bank))
    :precondition ((at ?person ?place) (at ?friend ?place) (drunk ?friend)
   (friendly ?friend ?person) (guard-of ?friend ?bank))
    :primitive t
    :text-template ("?Person lay ?friend to rest at ?place."
    (subj "?person"))
    :effect ((sleeping ?friend) (:not (guarded ?bank))))

;; take item off sleeping person
(define (action take-thing-off-sleeper)
```

```
    :parameters (?person ?sleeper ?thing ?place)
    :actors (?person)
    :constraints ((person ?person) (place ?place)
  (alley ?place) (person ?sleeper) (thing ?thing))
    :precondition ((at ?person ?place) (at ?sleeper ?place)
   (sleeping ?sleeper) (has ?sleeper ?thing))
    :primitive t
    :text-template ("?Person took ?thing off of ?sleeper."
    (subj "?person"))
    :effect ((has ?person ?thing)
     (:not (has ?sleeper ?thing))))

;; pawn a valuable for money
(define (action pawn-valuable)
    :parameters (?person ?pawn-broker ?thing ?place ?big-money)
    :actors (?person)
    :constraints ((person ?person) (pawn-broker ?pawn-broker)
  (big-money ?big-money)
  (place ?place) (thing ?thing) (valuable ?thing))
    :precondition ((at ?person ?place) (at ?pawn-broker ?place)
   (has ?person ?thing) (has ?pawn-broker ?big-money))
    :primitive t
    :text-template ("?Person pawned ?thing to ?pawn-broker for ?big-money."
    (subj "?person"))
    :effect ((:not (has ?person ?thing))
     (has ?pawn-broker ?thing)
     (has ?person ?big-money)))

;; buy a horse
(define (action buy-valuable)
    :parameters (?person ?seller ?thing ?place ?big-money)
    :actors (?person)
    :constraints ((person ?person) (seller ?seller)
  (big-money ?big-money)
  (place ?place) (thing ?thing) (valuable ?thing))
    :precondition ((has ?seller ?thing)
   (at ?person ?place) (at ?seller ?place)
   (has ?person ?big-money))
    :primitive t
    :text-template ("?Person bought ?thing from ?seller with ?big-money."
    (subj "?person"))
    :effect ((has ?person ?thing)
     (:not (has ?seller ?thing))
     (:not (has ?person ?big-money))))
```

```
;; ride a horse to a location
(define (action ride-horse-to)
    :parameters (?person ?horse ?from ?to)
    :actors (?person)
    :constraints ((mobile-person ?person) (place ?from) (place ?to)
  (connected ?from ?to) (horse ?horse))
    :precondition ((at ?person ?from) (at ?horse ?from)
   (has ?person ?horse))
    :primitive t
    :text-template ("?Person rode ?horse from ?from to ?to."
    (subj "?person"))
    :effect ((at ?person ?to) (:not (at ?person ?from))
     (at ?horse ?to) (:not (at ?horse ?from))))

;; hold up a bank
(define (action hold-up-bank)
    :parameters (?person ?gun ?bank ?sheriff)
    :actors (?person)
    :constraints ((person ?person) (evil ?person) (bank ?bank)
  (gun ?gun) (sheriff ?sheriff))
    :precondition ((at ?person ?bank) (has ?person ?gun)
   (:not (guarded ?bank)))
    :primitive t
    :text-template ("?Person held up ?bank with ?gun." (subj "?person"))
    :effect ((held-up ?person ?bank)
     (intends ?sheriff (arrested ?sheriff ?person))))

;; collect money
(define (action collect-money-from-heist)
    :parameters (?person ?bank ?mother-lode)
    :actors (?person)
    :constraints ((person ?person) (evil ?person) (bank ?bank)
  (mother-lode ?mother-lode))
    :precondition ((at ?person ?bank) (held-up ?person ?bank)
   (has ?bank ?mother-lode))
    :primitive t
    :text-template ("?Person collected ?mother-lode from ?bank."
    (subj "?person"))
    :effect ((has ?person ?mother-lode)
     (:not (held-up ?person ?bank))
     (:not (has ?bank ?mother-lode))))

;; getaway with stolen money
```

```
(define (action getaway-with-money)
    :parameters (?person ?mother-lode ?horse ?place ?dest)
    :actors (?person)
    :constraints ((person ?person) (evil ?person) (place ?place)
  (place ?dest) (mother-lode ?mother-lode) (horse ?horse)
  (connected ?place ?dest))
    :precondition ((at ?person ?place) (at ?horse ?place)
   (has ?person ?mother-lode))
    :primitive t
    :text-template ("?Person rode ?horse into the sunset with ?mother-lode."
    (subj "?person"))
    :effect ((:not (at ?person ?place))
     (:not (at ?horse ?place))
     (free-with-money ?person)
     (at ?person ?dest)))

;; bystanders sound alarm
(define (action alert-sheriff)
    :parameters (?person ?bystanders ?sheriff ?bank)
    :not-needs-intention t
    :constraints ((person ?person) (evil ?person) (sheriff ?sheriff)
  (bystanders ?bystanders) (bank ?bank))
    :precondition ((at ?person ?bank) (held-up ?person ?bank))
    :primitive t
    :text-template ("?Bystanders alerted ?sheriff that ?person was robbing the bank."
    (subj "?bystanders"))
    :effect ((knows ?sheriff (held-up ?person ?bank))))

(define (action arrest)
    :parameters (?criminal ?sheriff ?place ?cuffs ?money)
    :actors (?sheriff)
    :constraints ((person ?criminal) (evil ?criminal)
  (sheriff ?sheriff) (place ?place) (cuffs ?cuffs)
  (money ?money))
    :precondition ((at ?sheriff ?place) (at ?criminal ?place)
   (has ?sheriff ?cuffs) (has ?criminal ?money))
    :primitive t
    :text-template ("?sheriff arrested ?criminal and recovered the ?money."
    (subj "?sheriff"))
    :effect ((arrested ?sheriff ?criminal) (in-cuffs ?criminal ?cuffs)
     (has ?sheriff ?money) (:not (has ?criminal ?money))))
```

Planning Problem Definition:

```
(define (problem western)
    :inits (
    ;; characters
    (person Robbie)
    (person Tom)
    (person Sally)
    (person Barney)
    (person HorseSeller)
    (person PawnBroker)
    (person Jill)
    (person Anne)
    (person Child1)

    (mobile-person Robbie)
    (mobile-person Tom)
    (mobile-person Sally)
    (mobile-person Barney)

    ;; character attributes
    (evil Robbie)
    (sheriff Tom)
    (seller HorseSeller)
    (pawn-broker PawnBroker)
    (guard-of Barney Bank)
    (blocking Child1 DarkAlley)

    ;; places
    (place Bank)
    (place MainStreet)
    (place Saloon)
    (place DressShop)
    (place SheriffsOffice)
    (place SallysHome)
    (place DarkAlley)
    (place BarberShop)
    (place BarneysRoom)
    (place GeneralStore)
    (place Out-of-Town)

    ;; place attributes
    (bank Bank)
    (alley DarkAlley)
    (alley-of DarkAlley MainStreet)
```

```
(bar Saloon)
(store GeneralStore)

;; map
(connected Bank Mainstreet)
(connected Saloon Mainstreet)
(connected DressShop Mainstreet)
(connected SheriffsOffice Mainstreet)
(connected SallysHome Mainstreet)
(connected DarkAlley Mainstreet)

(connected Mainstreet Bank)
(connected Mainstreet Saloon)
(connected Mainstreet DressShop)
(connected Mainstreet SheriffsOffice)
(connected Mainstreet SallysHome)
(connected Mainstreet DarkAlley)

(connected SheriffsOffice BarberShop)
(connected BarberShop SheriffsOffice)
(connected BarneysRoom Saloon)
(connected BarberShop Out-of-Town)
(connected Bank Out-of-Town)

;; things
(thing MotherLode)
(thing SixShooter)
(thing DressMoney)
(thing LocketMoney)
(thing BrownHorse)
(thing WhiteHorse)
(thing Locket)
(thing HandCuffs)
(thing Tomatoes)
(thing BlueDress)
(poker-game PokerGame)

;; thing attributes
(gun SixShooter)
(big-money LocketMoney)
(horse BrownHorse)
(horse WhiteHorse)
(valuable Locket)
(cuffs HandCuffs)
```

```
 (small-goods Tomatoes)
 (money DressMoney)
 (money TomatoMoney)
 (money PokerMoney)
 (mother-lode MotherLode)
 (money MotherLode)
 (valuable BlueDress)

 ;; where the things are
 (has Bank MotherLode)
 (has Bank DressMoney)
 (at SixShooter BarneysRoom)
 (has HorseSeller BrownHorse)
 (at WhiteHorse BarberShop)
 (has PawnBroker LocketMoney)
 (has Robbie Locket)
 (at HandCuffs SheriffsOffice)
 (has Sally Tomatoes)
 (has Anne TomatoMoney)
 (forsale BlueDress GeneralStore)
 (at PokerGame Saloon)
 (bet-at PokerMoney PokerGame)

 ;; locations
 (at Robbie MainStreet)
 (at Sally MainStreet)
 (at Tom SheriffsOffice)
 (at Barney BarneysRoom)
 (at HorseSeller MainStreet)
 (at BrownHorse MainStreet)
 (at PawnBroker MainStreet)
 (at Jill GeneralStore)
 (at Anne MainStreet)

 ;; intentions
 (intends Robbie (has Robbie PokerMoney))
 (intends Sally (has Sally BlueDress))
 (intends Barney (at Barney Saloon))
 )

 :goal (
(arrested Tom Robbie)
)
 )
```

Step Listing (possible linearization):

```
[21]: (PICK-UP TOM HANDCUFFS SHERIFFSOFFICE)
[22]: (MOVE-ONCE TOM SHERIFFSOFFICE BARBERSHOP)
[19]: (HOLSTER-GUN BARNEY SIXSHOOTER BARNEYSROOM)
[20]: (MOVE-ONCE BARNEY BARNEYSROOM SALOON)
[26]: (WITHDRAW-MONEY SALLY BANK DRESSMONEY)
[28]: (OPEN JILL GENERALSTORE)
[29]: (MOVE-ONCE SALLY BANK MAINSTREET)
[27]: (SELL SALLY ANNE TOMATOES TOMATOMONEY MAINSTREET)
[18]: (HATCH-PLAN ROBBIE SIXSHOOTER BROWNHORSE BANK MOTHERLODE)
[34]: (KICK-OUT-OF-WAY ROBBIE CHILD1 MAINSTREET DARKALLEY)
[9]: (MOVE-ONCE ROBBIE MAINSTREET DARKALLEY)
[8]: (HIDE-IN-DARK-ALLEY ROBBIE DARKALLEY)
[7]: (PICKPOCKET ROBBIE SALLY DRESSMONEY MAINSTREET DARKALLEY)
[30]: (MOVE-ONCE SALLY MAINSTREET GENERALSTORE)
[31]: (FAIL-BUY-DRESS SALLY BLUEDRESS DRESSMONEY GENERALSTORE)
[32]: (BUY-DRESS SALLY BLUEDRESS TOMATOMONEY GENERALSTORE)
[10]: (MOVE-ONCE ROBBIE DARKALLEY MAINSTREET)
[11]: (MOVE-ONCE ROBBIE MAINSTREET SALOON)
[6]: (BUY-DRINKS-FOR ROBBIE BARNEY DRESSMONEY SALOON)
[12]: (ESCORT-DRUNK-FRIEND ROBBIE BARNEY SALOON MAINSTREET)
[13]: (ESCORT-DRUNK-FRIEND ROBBIE BARNEY MAINSTREET DARKALLEY)
[5]: (LAY-TO-REST-IN-ALLEY ROBBIE BARNEY DARKALLEY)
[4]: (TAKE-THING-OFF-SLEEPER ROBBIE BARNEY SIXSHOOTER DARKALLEY)
[14]: (MOVE-ONCE ROBBIE DARKALLEY MAINSTREET)
[24]: (MOVE-ONCE ROBBIE MAINSTREET SALOON)
[23]: (CHEAT-AT-POKER ROBBIE POKERGAME DRESSMONEY POKERMONEY SALOON)
[25]: (MOVE-ONCE ROBBIE SALOON BARBERSHOP)
[15]: (PAWN-VALUABLE ROBBIE PAWNBROKER LOCKET BARBERSHOP LOCKETMONEY)
[16]: (BUY-VALUABLE ROBBIE HORSESELLER BROWNHORSE BARBERSHOP LOCKETMONEY)
[17]: (RIDE-HORSE-TO ROBBIE BROWNHORSE MAINSTREET BANK)
[3]: (HOLD-UP-BANK ROBBIE SIXSHOOTER BANK)
[2]: (COLLECT-MONEY-FROM-HEIST ROBBIE BANK MOTHERLODE)
[1]: (GETAWAY-WITH-MONEY ROBBIE MOTHERLODE BROWNHORSE BANK OUT-OF-TOWN)
[33]: (RIDE-HORSE-TO TOM WHITEHORSE BARBERSHOP OUT-OF-TOWN)
[0]: (ARREST TOM ROBBIE HANDCUFFS MOTHERLODE OUT-OF-TOWN)
```

Intentions Listing:

```
ROBBIE intends (HAS ROBBIE MOTHERLODE) because (18)
   3: (HOLD-UP-BANK ROBBIE SIXSHOOTER BANK)
   2: (COLLECT-MONEY-FROM-HEIST ROBBIE BANK MOTHERLODE)
```

```
ROBBIE intends (HAS ROBBIE BROWNHORSE) because (18)
   14: (MOVE-ONCE ROBBIE DARKALLEY MAINSTREET)
   15: (PAWN-VALUABLE ROBBIE PAWNBROKER LOCKET MAINSTREET LOCKETMONEY)
   16: (BUY-VALUABLE ROBBIE HORSESELLER BROWNHORSE MAINSTREET LOCKETMONEY)
ROBBIE intends (HAS ROBBIE SIXSHOOTER) because (18)
   34: (KICK-OUT-OF-WAY ROBBIE CHILD1 MAINSTREET DARKALLEY)
   9: (MOVE-ONCE ROBBIE MAINSTREET DARKALLEY)
   8: (HIDE-IN-DARK-ALLEY ROBBIE DARKALLEY)
   7: (PICKPOCKET ROBBIE SALLY DRESSMONEY MAINSTREET DARKALLEY)
   10: (MOVE-ONCE ROBBIE DARKALLEY MAINSTREET)
   11: (MOVE-ONCE ROBBIE MAINSTREET SALOON)
   6: (BUY-DRINKS-FOR ROBBIE BARNEY DRESSMONEY SALOON)
   12: (ESCORT-DRUNK-FRIEND ROBBIE BARNEY SALOON MAINSTREET)
   13: (ESCORT-DRUNK-FRIEND ROBBIE BARNEY MAINSTREET DARKALLEY)
   5: (LAY-TO-REST-IN-ALLEY ROBBIE BARNEY DARKALLEY BANK)
   4: (TAKE-THING-OFF-SLEEPER ROBBIE BARNEY SIXSHOOTER DARKALLEY)
ROBBIE intends (FREE-WITH-MONEY ROBBIE) because (18)
   17: (RIDE-HORSE-TO ROBBIE BROWNHORSE MAINSTREET BANK)
   1: (GETAWAY-WITH-MONEY ROBBIE MOTHERLODE BROWNHORSE BANK)
ROBBIE intends (HAS ROBBIE POKERMONEY) because (INITIAL-STATE)
   14: (MOVE-ONCE ROBBIE DARKALLEY MAINSTREET)
   24: (MOVE-ONCE ROBBIE MAINSTREET SALOON)
   23: (CHEAT-AT-POKER ROBBIE POKERGAME DRESSMONEY POKERMONEY SALOON)
TOM intends (ARRESTED TOM ROBBIE) because (3)
   21: (PICK-UP TOM HANDCUFFS SHERIFFSOFFICE)
   22: (MOVE-ONCE TOM SHERIFFSOFFICE BARBERSHOP)
   33: (RIDE-HORSE-TO TOM WHITEHORSE BARBERSHOP OUT-OF-TOWN)
   0: (ARREST TOM ROBBIE HANDCUFFS MOTHERLODE OUT-OF-TOWN)
SALLY intends (HAS SALLY BLUEDRESS) because (INITIAL-STATE)
   26: (WITHDRAW-MONEY SALLY BANK DRESSMONEY)
   29: (MOVE-ONCE SALLY BANK MAINSTREET)
   27: (SELL SALLY ANNE TOMATOES TOMATOMONEY MAINSTREET)
   30: (MOVE-ONCE SALLY MAINSTREET GENERALSTORE)
   31: (FAIL-BUY-DRESS SALLY BLUEDRESS DRESSMONEY GENERALSTORE)
   32: (BUY-DRESS SALLY BLUEDRESS TOMATOMONEY GENERALSTORE)
BARNEY intends (AT BARNEY SALOON) because (INITIAL-STATE)
   19: (HOLSTER-GUN BARNEY SIXSHOOTER BARNEYSROOM)
   20: (MOVE-ONCE BARNEY BARNEYSROOM SALOON)
```

Story Texts

```
Fred was flat broke.
He kicked at a child on main street.
He slipped into a dark alley right next to main street.
Sally withdrew some money at the bank and put it in her purse.
She went from the bank to main street
Fred bought Barney a round of drinks.
```

Figure A.40: Story 1 (Causally Necessitated), Prompted

```
Fred was quite wealthy.
He kicked at a child on main street.
He slipped into a dark alley right next to main street.
Sally withdrew some money at the bank and put it in her purse.
She went from the bank to main street.
Fred bought Barney a round of drinks.
```

Figure A.41: Story 1 (Causally Necessitated), Un-prompted

```
Short Answer:
Where did Fred kick a child?

Word Recognition:
boot
alley
purse
drinks
beer
```

Figure A.42: Tests for for Story 1 (Causally Necessitated), Prompted and Un-prompted

```
Alex took his gun from his bedroom and put it into the holster on his hip.
Paul did not have a gun.
He escorted a drunken Alex into a dark alley.
Alex fell asleep.
Paul held up the bank with a gun.
```

Figure A.43: Story 2 (Causally Necessitated), Prompted

```
Alex took his gun from his bedroom and put it into the holster on his hip.
Paul had a gun.
He escorted a drunken Alex into a dark alley.
Alex fell asleep.
Paul held up the bank with a gun.
```

Figure A.44: Story 2 (Causally Necessitated), Un-prompted

```
Short Answer:
What did Paul do with a drunken Alex?

Word Recognition:
money
bank
asleep
holster
teller
```

Figure A.45: Tests for Story 2 (Causally Necessitated), Prompted and Un-prompted

```
Bob was the sheriff.
He wanted to arrest James.
He picked up his handcuffs at his office.
He went from his office to the barbershop.
James cheated at the poker game.
James went from the saloon to the barbershop.
```

Figure A.46: Story 3 (Intentionally Necessitated), Prompted

```
Bob was the sheriff.
He wanted to get a haircut.
He picked his handcuffs at his office.
He went from his office to the barbershop.
James cheated at the poker game.
James went from the saloon to the barbershop.
```

Figure A.47: Story 3 (Intentionally Necessitated), Un-prompted

```
Short Answer:
What did James do at the poker game?

Word Recognition:
sheriff
saloon
handcuffs
barbershop
razor
```

Figure A.48: Tests for Story 3 (Intentionally Necessitated), Prompted and Un-prompted

```
A blue dress with a long ribbon was on sale at the general store.
Cindy wanted something new to wear.
She withdrew some money from the bank.
She sold some tomatoes to Kelly for a little extra money.
Margery opened the general store for business.
Cindy went from main street to the general store.
```

Figure A.49: Story 4 (Intentionally Necessitated), Prompted

```
A blue dress with a long ribbon was on sale at the general store.
Cindy wanted something new to eat.
She withdrew some money from the bank.
She sold some tomatoes to Kelly for a little extra money.
Margery opened the general store for business.
Cindy went from main street to the general store.
```

Figure A.50: Story 4 (Intentionally Necessitated), Un-prompted

```
Short Answer
Why did Sally sell some tomatoes to Kelly?

Word Recognition
business
pants
ribbon
blue
dress
```

Figure A.51: Test for Story 4 (Intentionally Necessitated), Prompted and Un-prompted